



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ
ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ**


ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σύστημα καταχώρησης και διαχείρισης εργασιών δήμου

Δημητρόπουλος Αναστάσιος

Εισηγήτρια: Βελώνη Αναστασία

Η διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Βελώνη Αναστασία	Καθηγήτρια	
Έλληνας Ιωάννης	Καθηγητής	
Φατούρος Σταύρος	Αναπληρωτής Καθηγητής	

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε το διάστημα μεταξύ Ιουλίου 2019 και Μαρτίου 2020 στα πλαίσια του 5ετους προγράμματος σπουδών στο τμήμα Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής. Ως την ελάχιστη δυνατή μνεία, με την παρούσα παράγραφο οφείλω να ευχαριστήσω όλους όσους συνέβαλαν στην εκπόνησή της και ιδιαίτερα:

Την επιβλέπουσα καθηγήτρια μου, κα Βελώνη Αναστασία, για την πολύτιμη υποστήριξή της και το άριστο κλίμα συνεργασίας που διαμόρφωσε συμβάλλοντας τα μέγιστα για την κατάρτιση της διπλωματικής μου εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον πατέρα μου, Σπύρο Δημητρόπουλο, για τις αμέτρητες συμβουλές και υποδείξεις του στο παραγωγικό μέρος της εφαρμογής.

Ιδιαίτερα θερμές ευχαριστίες θέλω να δώσω στην οικογένεια μου για την συνεχείς συμπαράσταση της, για τις πολύτιμες συμβουλές και για όλα όσα μου έχουν προσφέρει όλα αυτά τα χρόνια της ζωής μου αλλά και των σπουδών μου.

Τέλος ευχαριστίες απευθύνω στην Φοιτήτρια του τμήματος Ιωάννα Βραχνάκη για όλη την υποστήριξη της, την υπομονή της και τις συμβουλές της.

Δημητρόπουλος Αναστάσιος
Αθήνα, Μάρτιος 2020

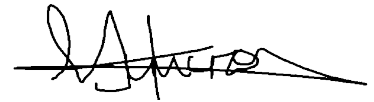
ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Δημητρώπουλος Αναστάσιος του Σπυρίδωνος, με αριθμό μητρώου 71343833 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Τεχνολογικών Εφαρμογών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα



Εισαγωγή	8
Περίληψη	8
Σκοπός	8
Τρόπος διευκόλυνσης κάτοικων του Δήμου	8
Abstract	9
Summary	9
Purpose	9
Κεφάλαιο 1 - Βάσεις Δεδομένων	10
Εισαγωγή	10
Κατηγορίες Βάσεων Δεδομένων	12
Οι Ιεραρχικές Βάσεις Δεδομένων	12
Οι Δικτυωτές Βάσεις Δεδομένων	13
Οι Σχεσιακές Βάσεις Δεδομένων	13
Ορισμός και Δομή των Βάσεων Δεδομένων	13
Ειδικοί της πληροφορικής και τεχνικές που χρησιμοποιούν	14
Σχέσεις Πινάκων	14
Σχέση Ένα-προς-Ένα(One-to-One)	15
Σχέση Πινάκων Ένα-προς-Πολλά (One-to-Many)	15
Σχέση Πολλά Προς Πολλά (Many-to-Many)	16
Κανονικοποίηση βάσεων δεδομένων	16
1 ^η Κανονική Μορφή (1NF-1 st Normal Form)	17
2 ^η Κανονική Μορφή (2NF-2 nd Normal Form)	19
3 ^η Κανονική Μορφή(3NF-3 rd Normal Form)	19
Ασφάλεια Βάσεων Δεδομένων	20
Έλεγχοι Διαχείρισης Βάσης Δεδομένων	21
Προληπτικοί έλεγχοι	22
Αποτροπή απώλειας δεδομένων (DLP-Data Loss prevention)	25
MySQL	26
Ταυτότητα της SQL	26
Ιστορία	26
Η γλώσσα SQL στη MySQL	27
Βασικές Εντολές της MySQL	28
Δημιουργία Βάσης Δεδομένων	28
Καταστροφή Βάσης Δεδομένων	29
Δημιουργία Πίνακα	29

Καταστροφή Πίνακα	29
Εισαγωγή Δεδομένων	29
Διαγραφή Δεδομένων	30
Τροποποίηση -Ενημέρωση Δεδομένων	30
Αναζήτηση Δεδομένων	30
Back-up Δεδομένων	31
Συναθροίσιμες Συναρτήσεις	31
Σχεσιακό Μοντέλο και Οντότητες	31
Ισχυρές Οντότητες	32
Ασθενείς Οντότητες	32
Περιορισμοί Ακεραιότητας	33
Σύνδεση Πινάκων	33
INNER JOIN	34
RIGHT JOIN	35
FULL JOIN	36
Κεφάλαιο 2 – Προγραμματισμός Ιστοσελίδων	38
Εισαγωγή	38
PHP	39
Ιστορία της PHP	40
Σύνταξη της PHP	40
Μεταβλητές στην PHP	41
Υπερκαθολικές Μεταβλητές	42
Η Ψευδοσυνάρτηση echo	43
JavaScript	45
Η Ιστορία της Javascript	45
Μοντέλο Εκτέλεσης	46
JQUERY	47
Χαρακτηριστικά	47
Χρήση της βιβλιοθήκης	47
Κεφάλαιο 3 – Σχεδιασμός Ιστοσελίδων	48
Σχεδιασμός Σελίδων Ιστού (Web Daring)	48
Προσβασιμότητα Ιστοσελίδων	50
Η Γλώσσα Σήμανσης HTML	51
Ορισμός	51
Ιστορία της Html	52
Πρώτες προδιαγραφές	52
Σήμανση	53
Στοιχεία	54
Ιδιότητες	57

Αναφορές οντοτήτων χαρακτήρων	58
Τύποι δεδομένων	59
Δήλωση τύπου εγγράφου	59
Σημασιολογική	59
Μεταφορά της HTML	60
HTTP	60
Ονόματα αρχείων	61
Δημιουργία με προγράμματα WYSIWYG	61
Φόρμες	61
CSS (Cascading Style Sheets)	63
Η αλληλουχία εφαρμογής των φύλλων ύφους	63
Κεφάλαιο 4- Προγραμματισμός με το PowerBuilder	65
Οι εφαρμογές PowerBuilder και ο χρήστης	65
Η γλώσσα PowerScript	66
Λειτουργίες PowerScript	66
Αντικειμενοστραφής προγραμματισμός με το PowerBuilder	66
.NET εφαρμογές	66
Πολλαπλές εφαρμογές	67
Σύνδεση βάσης δεδομένων	67
Ορισμός πηγής δεδομένων(data source)	67
Χρήση προφίλ βάσεων δεδομένων για σύνδεση	67
Συνδέσεις βάσης δεδομένων	68
Το περιβάλλον του PowerBuilder	68
Workspaces and εφαρμογές	68
Δέντρο συστήματος (System Tree)	69
Παράθυρο Clip	69
Παράθυρο εξόδου	70
Εργαλεία σχεδίασης	70
Έλεγχοι σχεδιασμού-χρόνου	70
Λίστα υποχρεώσεων(To do list)	70
Browser	70
PowerBar	71
PainterBar	71
PowerTips	71
Προσαρμογή του περιβάλλοντος	71
Αντικείμενα του PowerBuilder	72
Το αντικείμενο εφαρμογής	72
Αντικείμενα του DataWindow	74

Μορφές εμφάνισης, επεξεργασία στυλ και επικύρωση	75
Web DataWindow	76
Query	77
Δομές (Structures)	78
Αντικείμενα Χρήστη (Users Object)	78
Βιβλιοθήκες	79
Έργα	79
Κεφάλαιο 5 – Η εφαρμογή	80
Ο κωδικας στο αρχείο connection.php	80
Ο κώδικας στο αρχείο pass.php	80
Ο κώδικας το αρχείο main.php	82
Ο Κώδικας στο αρχείο success.php	84
Ο κώδικας στο αρχείο error.php	86
Ο κώδικας στο αρχείο passwordchange.php	88
Ο κώδικας στο αρχείο passwordcheck.php	93
Ο κώδικας στο αρχείο profile.php	95
Ο κώδικας στο αρχείο records.php	100
Ο Κώδικας στο αρχείο registration.php	103
Ο κώδικας στο αρχείο telephones.php	110
Ο κώδικας στο αρχείο update.php	118
Ο κωδικας στο αρχείο style.css	120
Ο Κώδικας στο αρχείο myscripts.js	123
Ο κωδικας του PowerBuilder	124
Η εφαρμογή Athens	124
Το παράθυρο w_pass	124
Ο κώδικας στο πλήκτρο 'ΕΙΣΟΔΟΣ'	125
Το παράθυρο w_main_menu	126
Το Πλήκτρο 'Καταχωρήσεις'	126
Το πλήκτρο Αναζήτησης Καταχώρησης	126
Το πλήκτρο Αναζήτηση Πελάτη	127
Το πλήκτρο Τηλεφωνα Δήμου	127
Το παραθυρο w_menu	127
Το dw_entries	128
Το πλήκτρο 'Ηλεκτρολογικό'	128
Το πλήκτρο 'Υδραυλικό'	129
Το πλήκτρο 'Κυκλοφοριακό'	129

To πλήκτρο 'Σήμανση'	130
To Drop Down List Box 'ddlb_katigories'	130
To πλήκτρο 'Προβολή'	131
To dw_histo	131
To παράθυρο w_ergasias	132
To Drop Down List Box 'ddlb_filter'	133
To dw_ergasias	133
To πλήκτρο 'cb_initialization'	133
To παράθυρο w_report_ergasias	135
To dw_report	136
To dw_dummy	137
To παραθυρο w_search	138
To πλήκτρο 'Προβολή'	138
To dw_anazitisi	139
To παραθυρο w_search_users	140
To dw_users	141
To dw_ergasias	141
To παραθυρο 'w_telephones	142
To Dw_telephones	142
To dw_dummy	142
Βιβλιογραφία	144

ΕΙΣΑΓΩΓΗ

Περίληψη

Σκοπός της εργασίας είναι η δημιουργία μια διαδικτυακής πλατφόρμας για την καταχώρηση και την διαχείριση τυχών βλαβών και συμβάντων στα πλαίσια ενός Δήμου καθώς επίσης και της άμεσης αλληλεπίδρασης των δημοτών με τον Δήμο. Συνεπώς, κύριος στόχος της πλατφόρμας είναι η εύκολη καταχώρηση μιας βλάβης στον δήμο από τους πολίτες, η αναβάθμιση των υπηρεσιών του δήμου και η σωστή οργάνωση και διαχείριση τυχόν εργασιών. Δόθηκε μεγάλη προσοχή στην απλότητα και στην ευκολία της πλατφόρμας τόσο από την πλευρά των πολιτών, όσο και από την πλευρά των διαχειριστών έτσι ώστε να είναι αρκετά εύκολη και αποτελεσματική και να αποτελεί αναπόσπαστο εργαλείο των πολιτών και των δημοσίων φορέων.

Σκοπός

Σκοπός της εργασίας είναι η δημιουργία μια πλατφόρμας για την καταχώρηση ,την διαχείριση τυχών βλαβών και συμβάντων καθώς επίσης και της άμεσης αλληλεπίδρασης των δημοτών με τον Δήμο. Συνεπώς, κύριος στόχος της πλατφόρμας είναι αναβάθμιση των υπηρεσιών του δήμου καθώς επίσης και η σωστή οργάνωση και διαχείριση τυχόν εργασιών.

Τρόπος διευκόλυνσης κάτοικων του Δήμου

Λαμβάνοντας υπόψη, ότι το διαδίκτυο αποτελεί ένα υπέρογκο φάσμα παροχής πληροφοριών, είναι εύκολο να αντιληφθεί κανείς ότι παρέχει απίστευτα οφέλη. Μέσω του διαδικτύου αναπτύσσονται και πιο πρακτικοί τομείς, οι οποίοι θα καταστήσουν ευκολότερη την καθημερινότητά των πολιτών, κάτι που μπορεί να γίνει μέσω των διαφόρων προγραμμάτων καταχώρησης, αρχειοθέτησης και ταξινόμησης πληροφοριών. Πλέον, η ευκολία να μπει κάποιος στο διαδίκτυο είναι πάρα πολύ μεγάλη καθώς όλοι διαθέτουμε 'έξυπνες' φορητές συσκευές (smartphones, tablet , κλπ.) καθώς επίσης και έναν ηλεκτρονικό υπολογιστή. Επιπλέον, η ενημέρωση του δήμου για κάθε συμβάν καθώς και η οργάνωση των δημοσίων υπηρεσιών είναι κάτι που φέρει περιθώρια βελτίωσης.

ABSTRACT

Summary

The purpose of the project is to create an online platform for the registration and management of any damages and incidents within a municipality, as well as the direct interaction of the citizens with the municipality. Therefore, the main objective of the platform is to easily record damage to the municipality by the citizens, to upgrade the municipal services and to properly organize and manage any work. Attention has been given to the simplicity and convenience of the platform on both the citizens' and the operators' side so that it is easy and effective enough and is an integral tool of citizens and public bodies.

Purpose

The purpose of the project is to create a platform for the recording, management of damages and incidents as well as the direct interaction of the citizens with the Municipality. Therefore, the main objective of the platform is to upgrade the municipal services as well as to properly organize and manage any work.

ΚΕΦΑΛΑΙΟ 1 - ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Εισαγωγή

Η ιστορία των βάσεων δεδομένων ξεκινάει περίπου 30 χρόνια πριν από σήμερα. Για την ακρίβεια, τα πρώτα δείγματα βάσεων δεδομένων, ή αλλιώς "τραπεζών δεδομένων" (data banks), όπως ονομάζονταν τότε, μπορούν να βρεθούν στα τέλη της δεκαετίας του '60. Εκείνη την εποχή τυποποιήθηκαν οι δικτυακές βάσεις δεδομένων (network databases), οι οποίες είναι γνωστές και με το όνομα βάσεις δεδομένων CODASYL, από το όνομα της επιτροπής που τις τυποποίησε. Επίσης, την εποχή εκείνη πρωτοεμφανίζεται και το IMS, το πιο γνωστό σύστημα διαχείρισης ιεραρχικών βάσεων δεδομένων. Το IMS, που είναι προϊόν της IBM, είναι ακόμα και σήμερα, ένα από τα συστήματα βάσεων δεδομένων με τις περισσότερες εγκαταστάσεις. Τα δύο μοντέλα αυτά χαρακτηρίζονται από το γεγονός ότι (α) η διαχείριση των δεδομένων γίνεται σε χαμηλό επίπεδο (με αποτέλεσμα όλες οι εφαρμογές να εξαρτώνται άμεσα από την τοποθεσία αποθήκευσης των δεδομένων) και (β) διαχειριζόμαστε τα δεδομένα ατομικά (record-at-a-time).

Το 1970, όμως, ο E. F. Codd δημοσιεύει το άρθρο "A Relational Model of Data for Large Shared Data Banks", στο περιοδικό "Communications of the ACM". Το εν λόγω άρθρο, θεωρείται ως η εργασία με τη μεγαλύτερη επίδραση στο χώρο των βάσεων δεδομένων, καθώς είναι αυτή που εισήγαγε το σχεσιακό μοντέλο βάσεων δεδομένων. Το σχεσιακό μοντέλο ήταν μια επαναστατική νέα στροφή στο νέο – για την εποχή εκείνη – χώρο των βάσεων δεδομένων. Το μοντέλο αυτό χαρακτηρίζεται από τη δυνατότητα μαζικής (set-at-a-time) επεξεργασίας δεδομένων και από την ανεξαρτησία των εφαρμογών από τη φυσική υλοποίηση. Το σχεσιακό μοντέλο, αφενός χαρακτηρίζεται από ένα πολύ ισχυρό θεωρητικό υπόβαθρο και αφετέρου από τη δυνατότητα άμεσης υλοποίησης: είναι μια από τις λίγες θεωρητικές εργασίες με τόσο μεγάλη πρακτική επίδραση. Ο E. F. Codd, μια δεκαετία αργότερα, το 1981, έλαβε το Turing Award για την συμβολή του αυτή.

Γύρω στα μέσα της δεκαετίας του '70, όμως, η μάχη κρίθηκε με την ανάπτυξη δύο σχεσιακών συστημάτων που είναι οι πρόγονοι όλων των εμπορικών συστημάτων σήμερα: της Ingres από το Πανεπιστήμιο της Καλιφόρνιας και του System R της IBM από το ερευνητικό κέντρο του San Jose (το οποίο τώρα ονομάζεται IBM Almaden Research Center). Η Ingres, κατασκευάστηκε από μια ομάδα ερευνητών με επικεφαλής τον M. Stonebraker. Σήμερα υπάρχουν διάφοροι απόγονοι της αρχικής αυτής προσπάθειας: Ingres, Sybase (και ως εκ τούτου και ο Microsoft SQL Server), Britton-

Lee/Sharebase, PACE. Ακόμα, σημαντική επιρροή είχε και στην αρχιτεκτονική της Informix. Το System R, από την άλλη πλευρά, φτιάχτηκε από ερευνητές της IBM (Irv Traiger, Bruce Lindsay, Paul McJones, Mike Blasgen, Mario Schkolnick, Bob Selinger, Bob Yost, Jim Gray), ήταν επίσης πρόγονος για διάφορα σχεσιακά συστήματα που υπάρχουν σήμερα: Oracle, HP Allbase, IBM SQL/DS, IBM DB2, Tandem Non-Stop SQL. Και τα δύο συστήματα, όντας ανταγωνιστικά μεταξύ τους, κυριάρχησαν στον εμπορικό, αλλά και τον ερευνητικό χώρο των σχεσιακών συστημάτων, μέχρι τις αρχές της δεκαετίας του '80, όταν η Oracle παίρνει τα πρωτεία από την IBM. Στα μέσα της δεκαετίας του '80, έχουμε το πρώτο στάνταρτ για μια γλώσσα διαχείρισης και επερώτησης σχεσιακών βάσεων δεδομένων, την Structured Query Language - SQL!

Στην σημερινή εποχή η τεχνολογία εξελίσσεται μέρα με την μέρα και η επιστήμη τόσο της πληροφορικής όσο και των επικοινωνιών είναι τεράστια. Συνεπώς η καθημερινότητα μας έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Μελέτες λένε ότι το αγαθό της "πληροφορίας" είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους. Μέσα από την "πληροφορία" καταφέρνουμε να είμαστε πιο αποδοτικοί, πιο ανταγωνιστικοί αλλά και πιο παραγωγικοί στην εργασία μας. Ένα σύστημα βάσεων δεδομένων το χρησιμοποιούμε για να μπορούμε να καταχωρήσουμε, να επεξεργαστούμε, αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

Με τον ορισμό βάση δεδομένων ορίζουμε λοιπόν μια οργανωμένη συλλογή από δεδομένα με σχήματα, πίνακες, ερωτήματα, εκθέσεις, προβολές και άλλα αντικείμενα. Επίσης ένας ακόμα ορισμός είναι ότι, μια βάση δεδομένων είναι ένα ολοκληρωμένο σύστημα που αποτελείται από δεδομένα (data) και από το κατάλληλο λογισμικό (software), τα οποία χρησιμοποιώντας το υλικό (hardware) βοηθούν στην ενημέρωση και πληροφόρηση των χρηστών (users). Τα δεδομένα συνήθως οργανώνονται για να μοντελοποιούν πτυχές της πραγματικότητας με τέτοιο τρόπο ώστε να μας παρέχουν τις απαραίτητες πληροφορίες.

(πηγές : Εισαγωγή στις Βάσεις Δεδομένων - ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ. Ν. ΦΛΩΡΙΝΑΣ ετ.2016

Π. Βασιλειάδης, Τ. Σελλής. Αντικειμενοστραφή Συστήματα Διαχείρισης Βάσεων Δεδομένων)

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

- ❖ **Πεδίο ή Field**, είναι το μικρότερο κομμάτι δεδομένων στο οποίο μπορούμε να αναφερθούμε και περιέχει ένα μόνο χαρακτηριστικό ή ιδιότητα ενός στοιχείου της βάσης δεδομένων.

- Το πεδίο αντιστοιχεί σε ένα δεδομένο και είναι το συστατικό στοιχείο μιας εγγραφής (record).
- Παραδείγματα πεδίων είναι ένα όνομα πελάτη, μια διεύθυνση, μια τιμή, ένας αριθμός τηλεφώνου.
- ❖ **Record ή Εγγραφή.** Ένα σύνολο διαφορετικών πεδίων που περιγράφουν ένα στοιχείο της βάσης ονομάζεται εγγραφή .
- ❖ **Πίνακας ή table.** Είναι ένα σύνολο από πολλά Records της βάσης δεδομένων.
- ❖ **Πρωτεύον Κλειδί ή Primary Key,** είναι ένα πεδίο που χαρακτηρίζει μοναδικά μια εγγραφή.
 - Οι εγγραφές αποθηκεύονται με βάση το πρωτεύον κλειδί.
 - Παράδειγμα πρωτεύοντος κλειδιού για τους φοιτητές του τμήματος είναι ο αριθμός μητρώου. Πολλές φορές η αυτόματη αρίθμηση του πίνακα της βάσης δεδομένων μπορεί από μόνη της να χρησιμοποιηθεί ως πρωτεύον κλειδί .
 - Το πρωτεύον κλειδί είναι μοναδικό για την κάθε εγγραφή ενός πίνακα και δεν είναι δυνατή η ύπαρξη δύο εγγραφών με το ίδιο κλειδί.
- ❖ **Ξένο Κλειδί ή Foreign Key,** χρησιμοποιείται για την διασύνδεση δυο ή περισσότερων πινάκων μεταξύ τους .

Οι στόχοι μιας βάσης δεδομένων είναι οι εξής :

- Η οργάνωση των δεδομένων και η εύκολη αναζήτηση αυτών όταν κριθεί αναγκαίο από τον εκάστοτε χρήστη.
- Ο καταμερισμός των ίδιων στοιχείων σ' όλους τους χρήστες.
- Η ομοιομορφία στον χειρισμό και την αναπαράσταση των δεδομένων.
- Η επιβολή κανόνων ασφαλείας (security).
- Η διατήρηση της ακεραιότητας και της αξιοπιστίας των δεδομένων(backup).

Κατηγορίες Βάσεων Δεδομένων

Οι Ιεραρχικές Βάσεις Δεδομένων

Στις Ιεραρχικές βάσεις δεδομένων τα δεδομένα αναπαρίστανται με δενδρικής μορφής δομές δεδομένων και συνδέονται μεταξύ τους με συνδέσμους (links). Η κάθε εγγραφή μπορεί να συνδέεται προς τα πάνω μόνο με μία άλλη εγγραφή (γονέας), ενώ μπορεί να έχει έως δύο εγγραφές που να εξαρτώνται απ' αυτήν (παιδιά). Υπάρχει μία μόνο εγγραφή ρίζα (root), απ' την οποία εξαρτώνται όλες οι άλλες εγγραφές της βάσης δεδομένων. Έχουν το μειονέκτημα ότι είναι πολύπλοκες στην επεξεργασία των εγγραφών τους (προσθήκη, διαγραφή, τροποποίηση).

Οι Δικτυωτές Βάσεις Δεδομένων

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

Οι Σχεσιακές Βάσεις Δεδομένων

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

(Πηγή :Η Θεωρία των Βάσεων Δεδομένων , Κέντρο ΠΛΗ.ΝΕ.Τ. ΦΛΩΡΙΝΑΣ)

Ορισμός και Δομή των Βάσεων Δεδομένων

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή πληροφοριών, το περιεχόμενο της οποίας μπορεί ευκολά και γρηγορά να:

- Προσπελασθεί
- Διαχειρισθεί
- Ενημερωθεί

Μια σχεσιακή βάση δεδομένων (Relational Database),κάνει χρήση της εννοίας των συνδεδεμένων διςδιάστατων πινάκων, που αποτελούνται από γραμμές και στήλες. Ένας τέτοιος πίνακας εμφανίζεται στο σχήμα 1.Ένας χρήστης μπορεί να χτίσει αναμεσα σε πολλαπλούς πίνακες και να παρουσιάσει το αποτέλεσμα του ως ένα νέο πίνακα.

Κωδικός Μαθήματος	Τίτλος μαθήματος	Εξάμηνο
2433493	Αρχιτεκτονική Υπολογιστών	4
4381232	Μαθηματική Ανάλυση	2
1237389	Δομές Δεδομένων	3

Σχήμα 1: Μορφή ενός τυπικού πίνακα μιας σχεσιακής βάσης δεδομένων.

Στη δομή αυτού του πίνακα οι γραμμές ονομάζονται εγγραφές (records) ενώ τα κελιά κάθε εγγραφής ονομάζονται ιδιότητες (Attributes) ή πεδία (Fields).Η

πληροφορία που χαρακτηρίζει και τακτοποιεί μοναδικά μια εγγραφή εντός του πίνακα, ονομάζεται κύριο κλειδί ή πρωτεύον κλειδί (primary key). Κάθε πίνακας έχει μόνο ένα πρωτεύον κλειδί. Ένα πρωτεύον κλειδί μπορεί να αποτελείται από ένα ή περισσότερα πεδία. Στην τελευταία περίπτωση, ένα τέτοιο πρωτεύον κλειδί ονομάζεται σύνθετο πρωτεύον κλειδί (Composite Primary Key). Στην πιο απλή τους μορφή, οι πίνακες είναι ανεξάρτητα αρχεία, με καλά οργανωμένη εσωτερική δομή. Μια βάση δεδομένων είναι ένας κατάλογος στον οποίο τοποθετούνται τα αρχεία-πίνακες. Κάποια επιπρόσθετα αρχεία υποστήριξης λειτουργιών μπορεί να βρίσκονται εντός της βάσης δεδομένων.

Ένας Database Server είναι ένα λογισμικό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS – Relational Database Management System) το οποίο κατά παρόμοιο τρόπο με έναν Web Server, μπορεί να περιέχει καταλόγους- βάσεις δεδομένων- τους οποίους προσφέρει προς χρήση στους χρήστες του μέσα από μια καλά καθορισμένη γλώσσα.

Μια δομημένη γλώσσα επερωτήσεων (SQL Structure Query Language) είναι σχεδόν μια φυσική γλώσσα, πολύ κοντά στην ανθρώπινη ιστορία της ξεκινά το 1971, όταν βασισμένοι σε μια εργασία του Dr. Edgar F. Codd πάνω σε σχεσιακά μοντέλα αποθήκευσης δεδομένων, οι ερευνητές της IBM Andrew Richardson, Donald C. Messerly και Raymond F. Boyce, δημιούργησαν μια απλή γλώσσα για χρήση με το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων System R της ίδιας εταιρίας, που ονόμαζαν SEQUEL (Structured English Query Language). Το 1986, η SQL πρωτυποποιήθηκε από το Αμερικανό Ινστιτούτο (ANSI – American National standards Institute) ως SQL-86, ενώ το 1987 από τον παγκόσμιο οργανισμό προτύπων (ISO International standards Organization).

Ειδικό της πληροφορικής και τεχνικές που χρησιμοποιούν

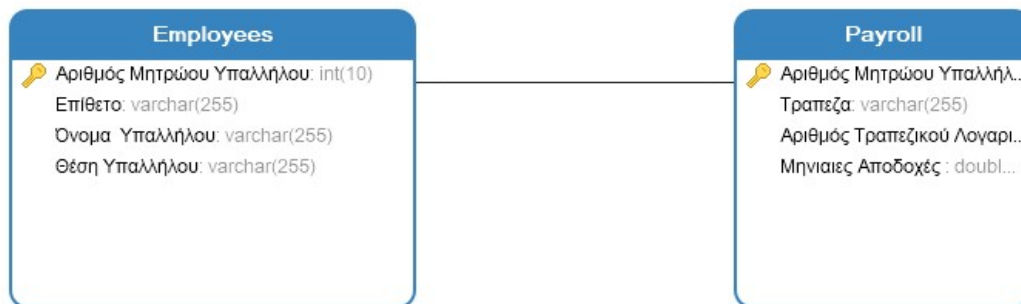
Για την ορθή ανάπτυξη και σωστή λειτουργία ενός συστήματος που βασίζεται στην τεχνολογία ατμών βάσεων δεδομένων χρειάζονται ειδικοί της πληροφορικής. Συγκεκριμένα, δυο κατηγορίες ειδικών της πληροφορικής είναι σημαντικές για την ανάπτυξη και λειτουργία μιας βάσης δεδομένων. Ο Σχεδιαστής και ο διαχειριστής της βάσης δεδομένων.

Σχέσεις Πινάκων

Μια σχεσιακή βάση δεδομένων, χτίζεται με έμφαση στον τρόπο που οι πίνακες συνδέονται μεταξύ τους, με σχέσεις αναμεσα στις εγγραφές τους. Αναγνωρίζονται συνολικά οι ακόλουθοι τρεις τύποι σχεσιακών πινάκων

Σχέση Ένα-προς-Ένα(One-to-One)

Στην σχέση 1 προς 1 μια εγγραφή του πίνακα σχετίζεται με 1 μόνο εγγραφή ενός άλλου πίνακα. Κλασικό παράδειγμα της σχέσης αυτής: έστω ότι στα πλαίσια μιας εταιρίας τα στοιχεία των υπάλληλων αποθηκεύονται στον Πίνακα Employees, ενώ στα στοιχεία μισθοδοσίας που τον αφορούν στον πίνακα Payroll. Ένας υπάλληλος μπορεί να έχει μόνο ένα κωδικό μισθοδοσίας ενώ ένας κωδικός αντιστοιχείται σε έναν και μόνο υπάλληλο. Με έμφαση στην ασφάλεια στον πρώτο πίνακα έχουν πρόσβαση όλοι οι υπάλληλοι, ενώ στο δεύτερο τμήμα έχουν πρόσβαση μόνο οι υπάλληλοι του λογιστηρίου

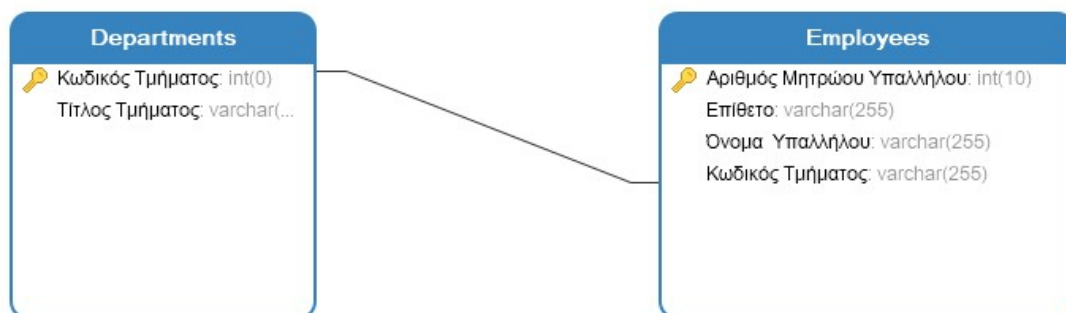


Σχήμα 2: Παράδειγμα σχέσεων πίνακα Ένα-προς-Ένα

Οι Δυο αυτοί πίνακες σχετίζονται μεταξύ τους με την χρήση πρωτευόντων κλειδιών τους (Αριθμός Μητρώου Υπάλληλου). Καθώς σε έναν πίνακα, το πρωτεύον κλειδί έχει εξ ορισμού μοναδική τιμή, εξασφαλίζεται ότι αναμεσα σε αυτούς τους πίνακες θα υπάρξει μια αντιστοιχία 1:1

Σχέση Πινάκων Ένα-προς-Πολλά (One-to-Many)

Στην σχέση ένα προς πολλά, μια εγγραφή του πίνακα σχετίζεται εγγραφές ενός ή πολλών πινάκων. Επεκτείνοντας το παράδειγμα της προηγούμενης σχέσης, μια εταιρία έχει αρκετά οργανωτικά τμήματα πχ. Λογιστήριο, ανθρωπίνων πόρων, πωλήσεων, διαφήμισης κλπ. Κάθε ένα τμήμα έχει πολλούς υπάλληλους, ενώ ένας υπάλληλος ανήκει σε έναν μόνο τμήμα

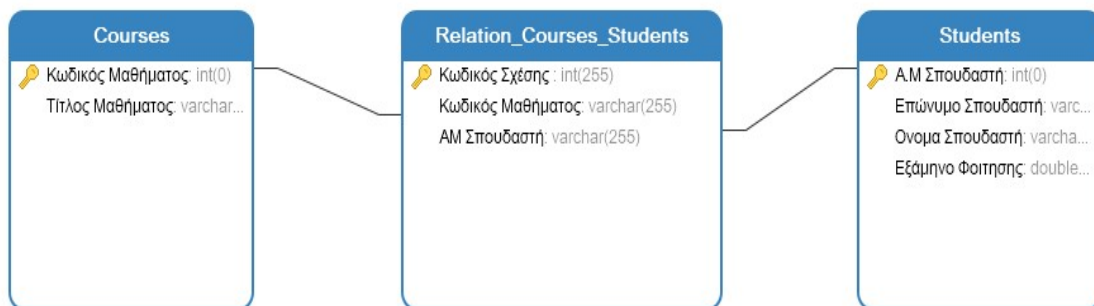


Σχήμα 3: Παράδειγμα σχέσεων πίνακα Ένα-προς-Πολλά

Ο πίνακας Departments περιέχει όλα τα τμήματα της εταιρίας, κάθε ένα με το δικό του μοναδικό πρωτεύον κλειδί. Στον πίνακα Employees έχει προστεθεί ένα πεδίο, που συνδέεται με το πρωτεύον κλειδί του departments και ονομάζεται ξένο κλειδί (foreign key) του πίνακα.

Σχέση Πολλά Προς Πολλά (Many-to-Many)

Στη σχέση Πολλά προς Πολλά, πολλές εγγραφές ενός πίνακα σχετίζονται με πολλές εγγραφές ενός ή περισσότερων πινάκων. Ως κλασσικό παράδειγμα, ένα μάθημα ενός εκπαιδευτικού ιδρύματος έχει πολλούς εγγεγραμμένους σπουδαστές που το παρακολουθούν. Επιπρόσθετα, οι ίδιοι σπουδαστές μπορούν να παρακολουθούν ο καθένας περισσότερα του ενός μαθήματα ταυτόχρονα σχέση αυτή υλοποιείται ως πολλαπλές σχέσεις ένα προς πολλά, χρησιμοποιώντας ενδιάμεσους πίνακες για την αποθήκευση των επιμέρους σχέσεων.



Σχήμα 4: Παράδειγμα σχέσεων πίνακα Πολλά-προς-Πολλά

Στο σχήμα 4, ο πίνακας Courses περιέχει όλα τα μαθήματα ενώ ο πίνακας students περιέχει τα προσωπικά στοιχεία κάθε σπουδαστή. Ο πίνακας Relation_Courses_Students ως εξωτερικά κλειδιά, τα πρωτεύοντα κλειδιά των άλλων δυο πινάκων. Οποιαδήποτε προσπάθεια καταρτιστεί ενός καταλόγου των σπουδαστών του παρακολουθούν ένα μάθημα, θα απαιτήσει την προσπέλαση και των τριών πινάκων, βάσει των σχέσεων τους. Αν ένας σπουδαστής δεν επιθυμεί να παρακολουθεί ένα μάθημα, αρκεί να αφαιρεθεί η αντίστοιχη σχέση από τον πίνακα Relation_Courses_Students, χωρίς να επηρεαστούν οι άλλοι δυο πίνακες.

Κανονικοποίηση βάσεων δεδομένων

Η κανονικοποίηση είναι η διαδικασία σύμφωνα με την οποία, μια βάση δεδομένων σχεδιάζεται κατά τέτοιο τρόπο, ώστε να αφαιρείται η πλεονάζουσα πληροφορία, να αυξάνεται η σαφήνεια της οργάνωσης και να μην παρεμποδίζεται η διαχείριση της.

Η κανονικοποίηση μια βάσης δεδομένων επιτρέπει μια ευκολότερη και αποδοτικότερη τροποποίηση της στο μέλλον, έτσι ώστε να καλύπτει τις προδιαγραφές μια συνεχώς αναβαθμιζομένης και λειτουργικά επεκτεινόμενης εφαρμογής.

Για τις ανάγκες της ενότητας, θα γίνει αναφορά στον ακόλουθο πίνακα Books_Full. Στον πίνακα αυτό, περιέχονται όλες οι σημαντικές πληροφορίες που αφορούν σε ένα βιβλίο.

ISBN	BookTitle	Author	Pages	Subject	Publisher
9784561	Βάσεις Δεδομένων	Τάσος Δημητρόπουλος	120	MySQL	Τάσος Δημητρόπουλος

Address	City	Prefecture	PoactalCode
Γλυκά Νερά	Αθήνα	Αθηνά	15354

Η δομή του πίνακα αυτού δεν είναι καθόλου αποδοτική. Για κάθε βιβλίο απαιτείται μια πλήρης εγγραφή, όπου τα στοιχεία του συγγραφέα και του εκδότη συχνά θα επαναλαμβάνονται στο πίνακα. Καθώς μάλιστα εισαγωγή των δεδομένων σε αυτό τον πίνακα θα γίνεται από ανθρώπους πιθανότητα παρουσίας σφαλμάτων ή διαφοροποιήσεων είναι πάρα πολύ μεγάλη. Επί πράγματι, σε μια εγγραφή βιβλίου το όνομα του εκδοτικού οίκου μπορεί να εισαχθεί από έναν υπάλληλο όπως ορίζουν πλήρως τα τιμολόγια ή σύμφωνα με κάποια δική του παράτυπη συντόμευση. Σε κάποια χρονική στιγμή εκδοτικός οίκος μπορεί να αλλάξει έδρα και οι παλαιές -πιθανότητα δεκάδες ή και εκατοντάδες βιβλία-θα πρέπει να διορθωθούν όλες με το χέρι. Αντίστοιχα ισχύουν και τα ονόματα των συγγραφέων, καθώς και για τις κατηγορίες στις οποίες κατατάσσεται ένα βιβλίο (πεδίο Subjects). Όλα αυτά τα επαναλαμβανόμενα πλεονάζοντα δεδομένα, απαιτούν σημαντικό αποθηκευτικό χώρο ο οποίος με αποδοτικότερη σχεδίαση της βάσης δεδομένων θα μπορούσε να ανακτηθεί. Επιπρόσθετα, η ύπαρξη πολλαπλών ονομάτων συγγραφέων σε ένα πεδίο καθιστά προβληματική την διαδικασία αναζήτησης και ταξινόμησης βάσει του συγγραφέα προσπάθεια ανεύρεσης όλων των βιβλίων που έχει γράψει ένας συγγραφέας και που ανήκουν σε κάποιο συγκεκριμένο είδος, απαιτεί ιδιαίτερες ικανότητες και συχνά μεγάλη υπομονή στην χρήση των σωστών ορών αναζήτησης.

Η εφαρμογή των ακολουθών τριών μορφών κανονικοποίησης των βάσεων δεδομένων, επιτυγχάνει πολλαπλά οφέλη για την εφαρμογή. Κάθε μια παρουσιαζόμενη μορφή κανονικοποίησης, απαιτεί την εφαρμογή των προηγούμενων

1^η Κανονική Μορφή (1NF-1st Normal Form)

Η 1^η κανονική μορφή, απαιτεί την σχεδίαση των πινάκων μια βάσης δεδομένων, κατά τέτοιον τρόπο ώστε κάθε πεδίο τους να περιέχει ένα κομμάτι δεδομένων (ατομικότητα). Ακολουθώντας τις σχέσεις πινάκων που παρουσιάστηκαν στην προηγούμενη ενότητα και επανασχεδιάζοντας τον πίνακα Books_Full σύμφωνα με την

μορφή αυτή, δημιουργούνται οι ακόλουθοι πίνακες. Τα πρωτεύοντα κλειδιά λαμβάνουν ως πρόθεμα του ονόματος τους το σύμβολο \diamond , ενώ τα εξωτερικά κλειδιά το σύμβολο \leftrightarrow

Πίνακας : Books

\diamond ISBN	BookTitle	Pages
9784561	Βάσεις Δεδομένων	120

Πίνακας : Book_Authors

\diamond AuthorID	AuthorLastName	AuthorFirstName
32	Δημητρόπουλος	Τάσος

Πίνακας: Book_Publishers

\diamond PublisherID	Publ_Name	Publ_Address	Publ_City	Publ_Prefecture	Publ_PostalCode
106	Δημητρόπουλος Αναστάσιος	Γλυκά Νερά	Αθήνα	Αθήνα	15354

Πίνακας:Book_Subjects

\diamond SubjectID	Subject_Caption
12	Software
54	Databases

Όπως γίνεται εμφανές, μετά την εφαρμογή της 1^{ης} κανονικής μορφής στον αρχικό πίνακα, από τον πίνακα που προέκυψε (Books), έχει αφαιρεθεί η πλεονάζουσα πληροφορία της εγγραφής του βιβλίου. Τα ονόματα των συγγραφέων εκδοτικός οίκος καθώς και οι κατηγορίες περιεχομένου, μεταφέρθηκαν σε ξεχωριστούς πίνακες (Book_Author, Book_Publishers και Book_Subject αντίστοιχα). Στα πρωτεύοντα κλειδιά των τριών τελευταίων πινάκων, έχουν δοθεί τυχαίες τιμές προκειμένου να δοθεί η εντύπωση πινάκων με περισσότερες εγγραφές και να μην υπάρξει σύγχυση μεταξύ πιθανά κοινών τιμών στην συνέχεια.

Πίνακας: Relation_Books_Author

\diamond Rel_BooksAuthorID	\leftrightarrow ISBN	\leftrightarrow AuthorID
45	9784561	32

Πίνακας: Relation_Books_Publisher

◇ Rel_BooksPublisherID	↔ISBN	↔PublisherID
16	9784561	106

Πίνακας: Relation_Books_Subjects

◇ Rel_BooksSubjectID	↔ISBN	↔Subject ID
8	9784561	12

2^η Κανονική Μορφή (2NF-2nd Normal Form)

Για την εφαρμογή της δεύτερης κανονικής εφαρμογής, οι πίνακες της βάσης δεδομένων, θα πρέπει να βρίσκονται ήδη στο 1^η κανονική μορφή και κάθε εγγραφή να έχει ένα πρωτεύον κλειδί. Κάθε πεδίο δεδομένων, πρέπει να είναι λειτουργικά εξαρτώμενο από ολόκληρο το πρωτεύον κλειδί. Όταν γίνεται χρήση συνθέτων πρωτευόντων κλειδίων, τα πεδία δεδομένων δεν μπορούν να εξαρτώνται από πεδίο-μέρος του πρωτεύοντος κλειδιού.

Πίνακας Book_Author_Comments

◇ ISBN	◇ AuthorID	Comment	AuthorEmail
9784561	32	Ένα βιβλίο για πρακτική εκμάθηση των βάσεων δεδομένων	cse43833@uniwa.gr

Ο πίνακας Book_Author_Comments περιέχει τα σχόλια κάθε συγγραμματος ενός βιβλίου. Τα πεδία ISBN και AuthorID αποτελούν το σύνθετο πρωτεύον κλειδί του πίνακα. Σε αυτόν τον πίνακα, το πεδίο AuthorEmail δεν εξαρτάται από όλα τα πεδία του συνθέτου πρωτεύοντος κλειδιού αλλά μόνο με το πεδίο AuthorID. Κατά συνέπεια, η πληροφορία του πεδίου AuthorEmail δεν ανήκει στον πίνακα αυτό. Η καταλληλότερη θέση για το πεδίο αυτό είναι ο πίνακας Books_Author, στο οποίο και θα πρέπει να μετακινηθεί.

3η Κανονική Μορφή (3NF-3rd Normal Form)

Για την εφαρμογή της 3^{ης} κανονικής μορφής, οι πίνακες της βάσης δεδομένων θα πρέπει να βρίσκονται ήδη σε 2^η κανονική μορφή. Κανένα πεδίο δεν πρέπει να εξαρτάται από την τιμή του άλλου πεδίου, πέρα από το ίδιο το πρωτεύον κλειδί του πίνακα. Αν σε τέτοιο πεδίο υπάρχει, αυτό θα πρέπει να τοποθετηθεί σε κάποιον άλλο πίνακα, και μια νέα σχέση να δημιουργηθεί ανάμεσα σε αυτόν και τον αρχικό πίνακα.

Στο παράδειγμα της ενότητας, ο πίνακας Book_Publishers περιέχει τρία πεδία δεδομένων τα οποία έχουν σχέσεις εξάρτησης μεταξύ τους. το πεδίο της πόλης

Publ_City εξαρτάται τόσο από το πεδίο του ταχυδρομικού κωδικό Publ_PostalCode, όσο και από το πεδίο της νομαρχίας Publ_Prefecture. Αν ο εκδοτικός οίκος μετακομίσει σε άλλη πόλη, στην εγγραφή του πίνακα Book_Publishers θα πρέπει να αλλαχθούν τουλάχιστον δυο από τα τρία αυτά πεδία εφαρμογή της κανονικής αυτής μορφής, θα οδηγήσει την τοποθέτηση των πεδίων της πόλης και της νομαρχίας σε δυο ξεχωριστούς πίνακες, ως ακολούθως. Το σχήμα αυτό επιτρέπει την μέγιστη ευελιξία της βάσης δεδομένων, αν και αυξάνει ίσως λίγο παραπάνω την πολυπλοκότητα της. Τα οφέλη είναι σημαντικά όταν γίνεται επαναχρησιμοποίηση των δεδομένων των δύο τελευταίων πινάκων.

PublisherID	Publ_Name	Publ_Address	Publ_PostalCode
106	Δημητρόπουλος Αναστάσιος	Γλυκά Νερά	15354

Ασφάλεια Βάσεων Δεδομένων

Με δεδομένη τη συνεχή αύξηση του όγκου της πληροφορίας των βάσεων δεδομένων, η προστασία αυτών καθίσταται μια ιδιαίτερα κρίσιμη λειτουργία μέσα στον Οργανισμό που απαιτεί συνεχείς ελέγχους με μια βηματική προσέγγιση στο επίπεδο διαχείρισης, πρόληψης και ανίχνευσης. Η ασφάλεια των βάσεων δεδομένων εξελίσσεται σε ένα ιδιαίτερος πολύπλοκο θέμα, το οποίο απαιτεί μια συστηματική προσέγγιση ώστε να καλυφθούν πολλές και διαφορετικές παράμετροι. Από τη μια μεριά διαπιστώνεται έξαρση στις επιλογές προστασίας, οι οποίες είναι τόσο ευρείας κάλυψης όσο και διεισδυτικές. Σαφώς όμως δεν υπάρχει μία λύση για κάθε περίπτωση και κατά συνέπεια απαιτούνται έλεγχοι που θα αποδεικνύουν ότι τα εταιρικά δεδομένα είναι όντως προφυλαγμένα με τον πιο αποτελεσματικό και αξιόπιστο τρόπο.

Από την άλλη μεριά, τα εταιρικά αυτά δεδομένα διαρκώς αυξάνουν σε όγκο – και μάλιστα με γοργό ρυθμό, κάτι που καθιστά σχεδόν αναγκαία τα ολοκληρωμένα συστήματα διαχείρισης των βάσεων δεδομένων (DBMS – database management system). Διαπιστώνεται όμως, ότι παρόλη την εξέλιξη που έχουν σημειώσει σε επίπεδο ασφάλειας, διατήρησαν έναν υψηλό βαθμό πολυπλοκότητας, που δεν καθιστά ευκρινή από τους διαχειριστές των βάσεων, την όποια αποτελεσματική προστασία προσφέρουν. Επίσης ο ρυθμός εξέλιξης των επιθέσεων στις βάσεις δεδομένων είναι τέτοιος, που τα DBMS δεν μπόρεσαν να ανταπεξέλθουν έγκαιρα, δημιουργώντας κενά ασφαλείας και την ανάγκη να αξιοποιηθούν εργαλεία τρίτων κατασκευαστών.

Με γνώμονα τα προαναφερόμενα θα αναφερθούν τρεις βασικές κατηγορίες ελέγχων που πρέπει να υιοθετηθούν από τους διαχειριστές των βάσεων δεδομένων, ώστε να υπάρχει ο μεγαλύτερος βαθμός προστασίας, εγρήγορσης και επίγνωσης σχετικά με τις δυνατότητες των χρησιμοποιούμενων μέτρων ασφαλείας. Οι έλεγχοι αναφέρονται σε χρήστες που διατηρούν σημαντικά επιχειρησιακά δεδομένα σε σχεσιακές βάσεις δεδομένων, αν και πλέον παρατηρείται και στοχοποίηση των μη σχεσιακών βάσεων δεδομένων, δημιουργώντας την ανάγκη αλλαγής προσέγγισης του ζητήματος της προστασίας τους.

Έλεγχοι Διαχείρισης Βάσης Δεδομένων

Διαχείριση της εγκατάστασης, μεταβολής και διαμόρφωσης μιας βάσης δεδομένων

Κατά την εγκατάσταση ενός DBMS είθισται να ακολουθείται ο εύκολος δρόμος και το θέμα της ασφάλειας να έρχεται σε δεύτερη μοίρα. Παρόλο που τα τελευταία χρόνια έχει αυξηθεί ο βαθμός εγρήγορσης στα θέματα προστασίας ακολουθούνται ακόμα επισφαλείς πρακτικές, όπως είναι οι κοινές ρυθμίσεις σε όλους τους λογαριασμούς των χρηστών, η εγκατάσταση του DBMS με ρυθμίσεις τυπικής διαμόρφωσης και η έκθεση των σεναρίων δειγματοληψίας και μερών του κώδικα που χρησιμοποιήθηκε.

Οι προμηθευτές των DBMS πλέον παρέχουν οδηγίες ασφάλειας και διαθέτουν σενάρια «ενδυνάμωσης» του συστήματος μετά την εγκατάσταση. Επιπρόσθετα υπάρχουν λύσεις από τρίτους κατασκευαστές, που καλύπτουν ευρύτερο φάσμα στο επίπεδο της προστασίας αν και τείνουν να είναι αυστηρότερες και επιβάλλουν την τέλεση εκτενών δοκιμών ώστε η εκάστοτε λύση να μην αποτελέσει πρόβλημα στην καθημερινή λειτουργία της εταιρείας.

Αφού επιλεγεί ο τρόπος προστασίας του DBMS και διαμορφωθεί όσο το δυνατόν αποτελεσματικότερα, ακολουθεί μια διαρκής και δια βίου διεργασία εκτίμησης των καθημερινών αλλαγών στο σύστημα και η εκ νέου διαμόρφωση των κανόνων. Στόχος των προαναφερόμενων διαδικασιών είναι η όσο το δυνατόν αρτιότερη προστασία του συνόλου του DBMS με την εισαγωγή του στο σύστημα διαχείρισης απειλών και τρωτών σημείων, χωρίς όμως να διαταράσσεται η ομαλή λειτουργία της εταιρείας.

Εντοπισμός και κατηγοριοποίηση των δεδομένων

Για να προστατευθούν ορθά τα δεδομένα ενός DBMS πρέπει να ξεκαθαριστεί αρχικά το είδος τους. Διαφορετικές ρυθμίσεις εφαρμόζονται για ευαίσθητα εταιρικά δεδομένα και άλλες για τους λογαριασμούς των χρηστών. Καθώς όμως ο όγκος τους αυξάνει με ρυθμό γεωμετρικής προόδου και οι πηγές προέλευσής τους είναι τόσο διαφορετικές, καθίσταται σχεδόν αδύνατο από ένα πρόγραμμα προστασίας δεδομένων να τα κατηγοριοποιήσει, να τους δώσει το σωστό βαθμό ευαισθησίας και να τα διασφαλίζει ικανοποιητικά.

Για την κατηγοριοποίηση των δεδομένων πρέπει να οριστούν εντός της εταιρείας τα αυτόνομα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS – Relational database management system) καθώς και αυτά που αποτελούν μέρος των εταιρικών συστημάτων. Ακολούθως να εντοπιστούν τα κρίσιμα δεδομένα εντός των βάσεων δεδομένων, αυτά δηλαδή που βάσει των πολιτικών ασφάλειας της εταιρείας και των νομικών απαιτήσεων χρήζουν προστασίας. Είθισται, όπου αυτό είναι δυνατό,

να εντάσσονται στο εταιρικό σύστημα ειδικά εργαλεία εντοπισμού των δεδομένων. Η πλήρης κατηγοριοποίηση και αξιολόγηση δεν είναι ποτέ εφικτή, αλλά με τους ανωτέρω τρόπους καλύπτεται σημαντικό εύρος.

Εντοπισμός τρωτών σημείων

Σε κάθε εταιρικό σύστημα υπάρχουν εργαλεία σάρωσης τα οποία εντοπίζουν τα τρωτά σημεία και τις απειλές που διαμορφώνονται σύμφωνα με τις εταιρικές απαιτήσεις. Ειδικότερα για τις βάσεις δεδομένων, αν και υπάρχουν εξειδικευμένα εργαλεία συνηθίζεται να χρησιμοποιούνται οι κοινοί σαρωτές δικτύου και εφαρμογών. Τα εργαλεία αυτά διαθέτουν ενεργητικούς και παθητικούς μηχανισμούς, με τους οποίους εντοπίζουν ήδη γνωστά τρωτά σημεία που είναι κοινά στα δικτυακά συστήματα, τυπικές αστοχίες διαμόρφωσης και άλλες αδυναμίες που προκύπτουν. Ο τρόπος που επιτυγχάνονται τα ανωτέρω είναι με τη στοχευμένη επίθεση στα εταιρικά συστήματα, την αποστολή επικίνδυνων πακέτων και την υποβάθμιση του λειτουργικού συστήματος και έτσι αξιολογούνται οι παράμετροι ασφαλείας που τέθηκαν. Οι διαχειριστές των βάσεων τελούν μία αρχική εκτίμηση για το αν το τρέχον εργαλείο σάρωσης της επιχείρησης μπορεί να αναβαθμιστεί και να περιλάβει στις διεργασίες του και τη βάση δεδομένων. Σε περίπτωση που χρησιμοποιούν ήδη κάποιο εξειδικευμένο σαρωτή για DBMS οφείλουν να το αξιολογούν διαρκώς ως προς την αξιοπιστία του, αλλά και τη δυνατότητά του να αποκαταστήσει και περαιτέρω τρωτά σημεία. Σε όλες τις περιπτώσεις τα ευρήματα των εργαλείων σάρωσης τρωτών σημείων είναι αυτά που καθοδηγούν τους διαχειριστές για τις απαιτούμενες αλλαγές και τη διαμόρφωση στόχων στο σύστημα.

Προληπτικοί έλεγχοι

Κρυπτογράφηση

Η κρυπτογράφηση αποτελεί ένα ισχυρό εργαλείο για την προστασία των δεδομένων – και ειδικά στις περιπτώσεις εναρμονισμού με κανόνες και νόμους είναι απαραίτητη. Παρόλα αυτά δεν αποτελεί την απόλυτη λύση και φυσικά παρουσιάζει περιορισμούς. Στις νεότερες εκδόσεις των εργαλείων κρυπτογράφησης, συχνά απαιτείται η κρυπτογράφηση της βάσης δεδομένων στο σύνολό της, με αρνητικές συνέπειες τόσο στην αποδοτική χρήση της όσο και στις εφαρμογές της επιχείρησης, οι οποίες μπορεί να παρακωλύονται από τα κρυπτογραφημένα δεδομένα. Προτού λοιπόν υιοθετηθεί ένα οποιοδήποτε εργαλείο κρυπτογράφησης, πρέπει να αξιολογηθεί εκτενώς ως προς την εφαρμοστικότητα του στο πραγματικό περιβάλλον λειτουργίας της επιχείρησης. Από την άλλη μεριά, η κρυπτογράφηση των δεδομένων δεν σημαίνει απαραίτητα ότι αυτά αποκρύπτονται από τους διαχειριστές της βάσης δεδομένων, που στην ουσία αποτελούν μέρος του προσωπικού και δεν θα έπρεπε να έχουν πρόσβαση στα κρίσιμα εταιρικά δεδομένα. Προς αυτή την κατεύθυνση τα νέα DBMS προσφέρουν δυνατότητες γηγενούς κρυπτογράφησης στο επίπεδο πινάκων και πεδίων της βάσης, ώστε οι διαχειριστές να μην έχουν πρόσβαση στα κλειδιά κρυπτογράφησης και στα

δεδομένα που αποθηκεύονται. Σε επίπεδο δεδομένων αναγνωρίζεται ότι η κρυπτογράφηση είναι ισχυρή μέθοδος προστασίας για το διαχωρισμό των καθηκόντων του κάθε χρήστη, για την απόκρυψη των δεδομένων από τους μη εξουσιοδοτημένους χρήστες και για την προστασία κατά τη μετακίνηση των δεδομένων, όπως συμβαίνει για παράδειγμα κατά τη δημιουργία αντιγράφων ασφαλείας. Αν όμως οι ρόλοι εντός της επιχείρησης δεν έχουν οριστεί με σαφήνεια, τότε όλες οι προσπάθειες κρυπτογράφησης μένουν στο κενό και τα κλειδιά κρυπτογράφησης μπορεί να βρεθούν εκτεθειμένα. Ένα υπό-τμήμα της κρυπτογράφησης αποτελεί η μέθοδος χρήσης token, με την οποία τα δεδομένα της βάσης αντικαθίστανται από tokens και τα πραγματικά δεδομένα απομακρύνονται από τη βάση, ειδικά σε περιπτώσεις όπου η πρόσβαση σε αυτήν είναι εύκολη. Η μέθοδος tokenization είναι ιδιαίτερος εξυπηρετική σε περιπτώσεις όπου τα δεδομένα είναι εύκολο να καθοριστούν, όπως είναι οι αριθμοί πιστωτικών καρτών ή το Α.Φ.Μ., αλλά δεν θεωρείται αποδοτική για πιο περίπλοκα δεδομένα όπως είναι οι ιατρικές πληροφορίες ενός ατόμου. Η χρήση της λοιπόν είναι περιορισμένη, αν και αποτελεσματική όπου το είδος των δεδομένων το επιτρέπει.

Συγκάλυψη δεδομένων

Η τεχνολογία της συγκάλυψης δεδομένων (data masking) εξυπηρετεί στη μη έκθεση των ευαίσθητων δεδομένων σε άτομα που απαιτείται να έχουν πρόσβαση σε αυτά, αλλά δεν είναι απαραίτητο να γνωρίζουν το ακριβές περιεχόμενό τους, όπως στο προσωπικό που τελεί δοκιμές ασφαλείας, στους προγραμματιστές και στους διαχειριστές των βάσεων. Η μέθοδος αντικαθιστά τα πραγματικά δεδομένα με άλλα εξίσου ρεαλιστικά ως προς τη μορφή τους, αλλά ψευδή ως προς το περιεχόμενό τους. Στη μέχρι τώρα μορφή της, η συγκάλυψη γινόταν στατικά και αφού τα δεδομένα είχαν αποθηκευτεί με στόχο την απόκρυψή τους από τους χρήστες, ενώ πλέον εισάγεται και η τεχνολογία της δυναμικής συγκάλυψης, όπου τα δεδομένα μεταβάλλονται σε πραγματικό χρόνο και χρησιμοποιείται στις βάσεις δεδομένων που συνδέονται άμεσα με τη γραμμή παραγωγής της επιχείρησης.

Διαχείριση πρόσβασης

Το σημείο κλειδί για την προστασία των εταιρικών δεδομένων από τους μη εξουσιοδοτημένους χρήστες είναι ο αυστηρός καθορισμός του ρόλου του κάθε υπαλλήλου και ο ορθολογικός σχεδιασμός των δικαιωμάτων του. Η ιδανική περίπτωση είναι ο κάθε υπάλληλος να έχει τα ελάχιστα δικαιώματα που του αναλογούν σε επίπεδο δικτύου, εφαρμογών και δεδομένων, τα οποία ταυτόχρονα θα του επιτρέπουν να τελεί απρόσκοπτα την εργασία του, αν και είναι δύσκολο να επιτευχθεί. Τα πιο κοινά τρωτά σημεία κατά τον ορισμό της προσβασιμότητας στις βάσεις δεδομένων, είναι:

- ❖ Η άμεση πρόσβαση στη βάση δίχως να παρεμβάλλεται κάποια εφαρμογή, η οποία συνήθως έχει κανόνες. Έτσι παραγκωνίζεται ο έλεγχος των εφαρμογών και ο χρήστης λειτουργεί κατά το δοκούν.
- ❖ Η χρήση από τους διαχειριστές όλων των δικαιωμάτων πρόσβασης που έχουν ακόμα και για να τελέσουν απλές καθημερινές διεργασίες, κινδυνεύοντας έτσι

να εκθέσουν πολύτιμα δεδομένα της βάσης.

- ❖ Η ελλιπής καταγραφή των αναγκαίων ρόλων που απαιτεί η επιχείρηση και άρα η κακή διαχείριση της πρόσβασης.

Η τέλεση ελέγχων πρόσβασης μόνο σε ένα επίπεδο π.χ. εφαρμογών και όχι στο σύνολο του εταιρικού συστήματος.

Για να αντιμετωπιστούν τα προαναφερόμενα προτείνεται το δίκτυο της εταιρείας να διαχωριστεί λογικά και να εφαρμοστεί κλιμακωτή αρχιτεκτονική. Βάσει αυτών να δημιουργηθούν λίστες πρόσβασης για κάθε τμήμα και ανάλογα το είδος του έργου που παράγει ο κάθε υπάλληλος, ώστε να αποφευχθεί η παράνομη πρόσβαση στις βάσεις δεδομένων της εταιρείας, ενώ ο έλεγχος πρόσβασης θα πρέπει να τελείται σε επίπεδο βάσης και όχι μόνο σε επίπεδο εφαρμογών. Επίσης αποτελεσματική θεωρείται η χρήση των βάσεων δεδομένων μέσω εικονικών μηχανών (jump box) ώστε να τελούνται οι ενέργειες των διαχειριστών, διατηρώντας όμως περιορισμένη πρόσβαση στα εργαλεία διαμόρφωσης της βάσης, όπως είναι το Toad και το SQLTools.

Έλεγχοι ανίχνευσης

Η προσπάθεια διασφάλισης των βάσεων δεδομένων σε βάθος είναι περίπλοκη διαδικασία και διαπιστώνεται ότι οι Οργανισμοί αγωνίζονται για να την επιτύχουν. Ακόμα και αν εφαρμοστούν όλες οι ανωτέρω μέθοδοι με τον καλύτερο δυνατό τρόπο, ελλοχεύουν κίνδυνοι για τα δεδομένα. Αποδεχόμενοι ότι τα δεδομένα έχουν αξία εφόσον χρησιμοποιούνται, γίνεται αντιληπτό ότι η χρήση τους περιλαμβάνει ρίσκο. Ακόμα και στις περιπτώσεις όπου ο χρήστης είναι εξουσιοδοτημένος, καμία μέθοδος που αναφέρθηκε δεν καλύπτει το γεγονός ότι μπορεί να είναι κακόβουλος. Από την άλλη μεριά, ο ισχυρός έλεγχος της προσβασιμότητας είναι αποδοτικός αλλά σχεδόν ανέφικτος σε πολύπλοκες δομές και εκτενή εταιρικά συστήματα. Γι' αυτόν το λόγο απαιτείται η υιοθέτηση ελέγχων ανίχνευσης, οι οποίοι συμπληρώνουν τα κενά ασφαλείας που παρουσιάζονται στους άλλους ελέγχους και προλαμβάνουν την ακατάλληλη χρήση των δεδομένων.

Παρακολούθηση της δραστηριότητας χρήσης των δεδομένων

Ο έλεγχος της δραστηριότητας των δεδομένων έχει ως στόχο την αποκάλυψη παράνομων ή ακατάλληλων ενεργειών από τους χρήστες, χωρίς να παρακωλύεται με κανέναν τρόπο η χρήση των δεδομένων για τις εφαρμογές της επιχείρησης. Είτε ως πρόσθετο στο κομμάτι της ασφάλειας των DBMS είτε ως ξεχωριστό εργαλείο, η ανίχνευση χρήσης επικεντρώνεται σε δύο βασικούς στόχους:

- ❖ παρακολούθηση των χρηστών με υψηλή προσβασιμότητα, ανάλυση και σύνταξη αναφορών της δραστηριότητάς τους

- ❖ παρακολούθηση των χρηστών των εφαρμογών και καταγραφή των γεγονότων πρόσβασης τους στις βάσεις δεδομένων.

Με αυτόν τον τρόπο μπορούν να τεθούν όρια χρήσης των δεδομένων και να υπάρχει καλύτερη επίβλεψη των υπαλλήλων και των ενεργειών τους, που από το σύστημα θεωρούνται ύποπτες.

Αποτροπή απώλειας δεδομένων (DLP-Data Loss prevention)

Τα DLP εργαλεία επικεντρώνονται αυστηρά στα δεδομένα των βάσεων, τα αξιολογούν με βάση την πολιτική ασφάλειας της εταιρείας και θέτουν τους κανόνες για την παρακολούθησή τους και την αποτροπή της απώλειάς τους, ενώ διαθέτουν δυνατότητες αποκατάστασης. Στόχος τους είναι η μη έκθεση των δεδομένων λόγω ακατάλληλης χρήσης τους από έναν αδαή υπάλληλο. Η χρήση τους στις βάσεις δεδομένων έχει πολλούς περιορισμούς, μιας και πραγματεύονται μόνο τα ίδια τα δεδομένα και δεν συνεργάζονται με την SQL, που είναι ο μηχανισμός επικοινωνίας με τα RDBMS.

Πληροφορίες ασφάλειας και διαχείριση συμβάντων (SIEM)

Η τεχνολογία SIEM (Security Information and Event Management) διαθέτει δύο βασικές λειτουργίες προστασίας των δεδομένων:

- ❖ Συλλογή, σύνταξη αναφορών και ανάλυση των αρχείων καταγραφής συμβάντων
- ❖ Διαχείριση ασφάλειας με την οποία συλλέγει δεδομένα από τις συσκευές ασφαλείας της εταιρείας, από τις δικτυακές συσκευές, από τα συστήματα και τις εφαρμογές σε πραγματικό χρόνο, τελεί συσχετισμούς μεταξύ των καταγεγραμμένων πληροφοριών και αναφέρει τυχόν συμβάντα ασφάλειας.

Η ενσωμάτωση των βάσεων δεδομένων στα συστήματα SIEM αν και πρακτική είναι επίπονη διαδικασία, μιας και υπάρχει και εδώ το πρόβλημα της SQL επικοινωνίας.

MySQL

Ταυτότητα της SQL

Η MySQL είναι μια πολύ γρήγορη σε απόδοση ,και ισχυρή σε δυνατότητες υλοποίησης ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων(RDBMS).Για τον χειρισμό των βάσεων και των πινάκων τους, στηρίζεται στην γλώσσα SQL,και συγκεκριμένα στην προτυποποιημένη της έκδοση SQL-92.Η ευκολία στην εγκατάσταση, ρύθμιση και χρήση, την καθιστούν μια ιδιαίτερα φιλική στην χρήστη - προγραμματιστή πλατφόρμα για την ανάπτυξη εφαρμογών.

Η υψηλή απόδοση της MySQL σε απαιτητικές εμπορικές εφαρμογές, κίνησε το ενδιαφέρον ενός από τους μεγαλύτερους κατασκευαστές λογισμικού για Servers,της SUN Microsystems(επίσης κατασκευάστρια της γλώσσας προγραμματισμού JAVA),η οποία και την εξαγόρασε το 2008.Η SUN Microsystems εξαγαράσαστηκε με την σειρά της τον Απρίλιο του 2000 από το μεγαλύτερο κατασκευαστή λογισμικού για βάσεις δεδομένων παγκοσμίως ,την Oracle.Σύμφωνα με την τελευταία, ένα από τους σκοπούς της εξαγοράς αυτής, ήταν η ενσωμάτωση της τεχνολογίας της MySQL ΣΤΑ ήδη καθιερωμένα προϊόντα της (Oracle SQL Server)

Η MYSQL ξεκίνησε την επιτυχημένη και ακόμα πολλά υποσχόμενη ζωή της, ως λογισμικό ανοιχτού κώδικα (Open Source Software).Κατά επέκταση, είναι ελεύθερη προς χρήση χωρίς κόστος απόκτησης για τις βασικές εκδόσεις της, οι οποίες όμως είναι πλήρως λειτουργικές. Καθώς ο πηγαίος της κώδικας είναι ελευθέρως διαθέσιμος, δυαδικές υλοποιήσεις έχουν παρουσιαστεί για περισσότερες από 20 διαφορετικές υπολογιστές πλατφόρμες, με τις δημοφιλέστερες να είναι αυτές που υποστηρίζουν στα λειτουργικά συστήματα Windows και Unix.

Ιστορία

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (Server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle. Η αρχική ανάπτυξη της MySQL από τους Widenius και Axmark ξεκίνησε το 1994. Η πρώτη έκδοση του MySQL εμφανίστηκε στις 23 Μαΐου 1995.Αρχικά δημιουργήθηκε για προσωπική χρήση από το mSQL με βάση το ISAM χαμηλού επιπέδου γλώσσα, το οποίο οι δημιουργοί θεωρούσαν πολύ αργούς και άκαμπτες. Δημιούργησαν μια νέα διεπαφή SQL, διατηρώντας παράλληλα το ίδιο API με το m SQL. Διατηρώντας το API συνεπές με το σύστημα mSQL, πολλοί

προγραμματιστές μπόρεσαν να χρησιμοποιήσουν την MySQL αντί για το προηγούμενο mSQL (ιδιοκτήτης).

Η γλώσσα SQL στη MySQL

Η γλώσσα επερωτησάντων SQL-92 με τις προσθήκες-επεκτάσεις της MySQL, διαίρεται σε αρκετά διαφορετικά στοιχεία. Σε αυτά ξεχωρίζουν οι ακόλουθες γλώσσες και κατηγορίες εντολών:

Data Description Language (DDL) : Γλώσσα περιγραφής δεδομένων, περιέχει τις εντολές που επιτρέπουν τη δημιουργία πινάκων, την τροποποίηση και τη διαγραφή τους. Χαρακτηριστικές εντολές είναι οι CREATE, ALTER ΚΑΙ DROP.

Data Manipulation Language (DML): Γλώσσα χειρισμού δεδομένων. Περιέχει τις εντολές που επιτρέπουν την εισαγωγή εγγράφων σε πίνακα, την τροποποίηση και την διαγραφή τους. Χαρακτηριστικές εντολές της γλώσσας αυτής είναι οι INSERT, UPDATE, DELETE και REPLACE

Queries: Τα ερωτήματα χρησιμοποιούνται για την ανάκτηση εγγράφων από ένα ή περισσότερους πίνακες, βάσει των σχηματισθέντων σχέσεων τους. Καθώς η εκτέλεση των ερωτημάτων δεν επηρεάζει τα αποθηκευμένα δεδομένα της βάσης, δεν μπορούν να θεωρηθούν ως μέρος γλώσσας χειρισμού δεδομένων.

Data Control Language (DCL): Γλώσσα ελέγχου δεδομένων. Περιέχει τις εντολές που ελέγχουν τα δικαιώματα πρόσβασης των χρηστών στις βάσεις δεδομένων και στους πίνακες τους. Χαρακτηριστικές εντολές της γλώσσας αυτής είναι οι GRANT και REVOKE.

Transaction Control: Η κατηγορία αυτή, περιέχει τις εντολές START TRANSACTION, COMMIT και ROLLBACK, που χρησιμοποιούνται για την εκτέλεση συναλλαγών (Transactions) με την βάση δεδομένων. Η MySQL υποστηρίζει τις συναλλαγές για πίνακες τύπου INNODB ή BerkleyDB (BDB). Ο εξ' ορισμού χρησιμοποιημένος τύπος δομής πινάκων της MySQL, MyISAM, δεν υποστηρίζει συναλλαγές.

Τα παραπάνω στοιχεία θα παρουσιαστούν αναλυτικά και παρακάτω. Όλες οι εντολές πρέπει να τερματίζονται με τον χαρακτήρα ";".

Κατηγορίες Δεδομένων

Όλα τα πεδία των πινάκων μπορούν να αποθηκεύσουν προκαθορισμένο τύπο δεδομένων. Ορισμένοι από τους υποστηριζόμενους τύπους δεδομένων είναι οι παρακάτω:

- ❖ **INT**. Ακέραιος αριθμός μήκους 3 bytes(24 bit).Δέχεται τιμές από το σύνολο [-32,768,...,32,767].Σε μη προσημασμένη μορφή δέχεται τιμές από το σύνολο [0...16,777,215]
- ❖ **BOOLEAN**. Αντιπροσωπεύει μεταβλητές με λογικές τιμές True ή False.
- ❖ **VARCHAR**.Δέσμευση μεγίστου πλήθους X χαρακτήρων (bytes).Το μέγιστο μήκος διαφέρει ανάλογα με την εκάστοτε βάση που χρησιμοποιείται. Μέγιστο μήκος συμβολοσειράς 255 χαρακτήρες
- ❖ **DOUBLE**. Αντιπροσωπεύει μεγάλο αριθμό κινητής υποδιαστολής
- ❖ **TEXT**.Κείμενο με μέγιστο όριο τους 65,535 χαρακτήρες
- ❖ **NULL**. Η σταθερά αυτή ταυτίζεται με την έννοια της ‘μη αποθηκευμένης τιμής’. Ένα κενό από τιμή πεδίο θεωρείται ότι περιέχει την τιμή NULL.
- ❖ **YEAR**. Χρησιμοποιείται για την τετραψήφια αναπαράσταση έτους
- ❖ **DATE**. Χρησιμοποιείται για καταχώρηση ημερομηνίας

Βασικές Εντολές της MySQL

Τα εργαλεία χειρισμού πληροφοριών σε μία βάση δεδομένων είναι γνωστά και ως γλώσσες εντολών. Μέσω αυτών έχουμε την δυνατότητα να δώσουμε εντολές χειρισμού δεδομένων. Οι κυριότερες εντολές της MySQL είναι εξής:

Δημιουργία Βάσης Δεδομένων

Έτσι, στο SQL το πρώτο βήμα για την αποθήκευση των δεδομένων με καλά δομημένο τρόπο είναι η δημιουργία μιας βάσης δεδομένων. Η εντολή CREATE DATABASE χρησιμοποιείται για τη δημιουργία μιας νέας βάσης δεδομένων στο SQL.

Με την πιο απλή εντολή της CREATE DATABASE:

- ❖ Δημιουργείται μια βάση δεδομένων.
- ❖ Δίνεται όνομα σε αυτή.

Παράδειγμα:

```
CREATE DATABASE athens;
```

Καταστροφή Βάσης Δεδομένων

Μια βάση δεδομένων που δημιουργήσαμε με την εντολή CREATE DATABASE , μπορούμε να την διαγράψουμε μέσω της εντολής DROP DATABASE .Η τελευταία συντάσσεται ως εξής:

```
DROP DATABASE athens;
```

Δημιουργία Πίνακα

Με την πιο απλή εντολή της create table :

- ❖ Δημιουργείται ένας πίνακας.
- ❖ Δίνεται όνομα σε αυτόν.
- ❖ Δηλώνονται τα ονόματα και οι τύποι δεδομένων των στηλών που περιέχει ο πίνακας.

Παράδειγμα:

```
CREATE TABLE users (id int NOT NULL AUTO_INCREMENT,name
varchar(255),surname varchar(255),email varchar(255),cellphone
varchar(10),password varchar(255), PRIMARY KEY (id));
```

Καταστροφή Πίνακα

Ένας πίνακα που δημιουργήσαμε με την εντολή CREATE TABLE , μπορούμε να τον διαγράψουμε μέσω της εντολής DROP TABLE .Η τελευταία συντάσσεται ως εξής:

```
DROP TABLE users;
```

Εισαγωγή Δεδομένων

Η εισαγωγή δεδομένων στην SQL γίνεται με την εντολή INSERT , η οποία εισάγει σε έναν πίνακα μια ολόκληρη εγγραφή. Το συντακτικό της εντολής INSERT είναι :

Παράδειγμα:

```
INSERT INTO users VALUES (0, 'Τάσος', 'Δημητρόπουλος', 'tdimitro17@gmail.com',
'6971234567', '43833');
```

Παρατηρήσεις :

- ❖ Η λίστα τιμών περιλαμβάνει όλα τα δεδομένα που θα εισαχθούν στον πίνακα, χωρισμένα με κόμμα (,).
- ❖ Το πλήθος των στηλών προς καταχώρηση πρέπει να είναι ίδιο με το πλήθος το πλήθος τις λίστας τιμών του πίνακα.

- ❖ Η σειρά με την οποία εισάγονται τα δεδομένα στις στήλες του πίνακα ,είναι αντίστοιχη με την σειρά που δηλώνονται οι τιμές στην λίστα τιμών.
- ❖ Οι τύποι των δεδομένων προς καταχώρηση πρέπει να συμφωνούν με την λίστα των στηλών του πίνακα αντίστοιχα.

Παράδειγμα :

```
"INSERT INTO users (name,surname,email,cellphone,password) VALUES ('Τάσος','Δημητρόπουλος','cse43833@uniwa.gr','6971234567','43833')";
```

Διαγραφή Δεδομένων

Για την διαγραφή δεδομένων από έναν πίνακα χρησιμοποιούμε την εντολή DELETE η οποία συντάσσεται σε δυο μορφές:

- ❖ Την απλή μορφή της η οποία δεν περιλαμβάνει την τιμή WHERE.Στην μορφή αυτή η εντολή διαγράφει όλες τις εγγραφές του πίνακα
- ❖ Στην σύνθετη μορφή της η εντολή περιέχει και την γραμμή WHERE οπου τότε θα διαγράψει μόνο τις εγγραφές του πίνακα που ικανοποιούν την συνθήκη.

Παράδειγμα:

```
DELETE FROM users WHERE email='cse43833@uniwa.gr';
```

Τροποποίηση -Ενημέρωση Δεδομένων

Η τροποποίηση των δεδομένων που υπάρχουν στην βάση μας , σε έναν πίνακα, γίνεται με την χρήση της εντολής UPDATE , η οποία έχει το εξής συντακτικό :

Παράδειγμα:

```
UPDATE users SET password =12345 WHERE email='cse43833@uniwa.gr';
```

Όπως η DELETE έτσι και η UPDATE παρατηρούμε ότι έχουν δυο μορφές :

- ❖ Στην απλή μορφή τις δεν περιλαμβάνει την γραμμή WHERE οπότε και η εντολή τροποποιεί όλες τις εγγραφές του πίνακα
- ❖ Στην σύνθετη μορφή της όταν περιέχει την τιμή WHERE η εντολή θα τροποποιήσει μόνο τις εγγραφές του πίνακα που ικανοποιούν την συνθήκη .

Αναζήτηση Δεδομένων

Η αναζήτηση των δεδομένων που έχουν αποθηκευτεί σε μία βάση δεδομένων και η ταυτόχρονη εμφάνιση τους σε μορφή πίνακα γίνεται με χρήση της εντολής SELECT. Η εντολή αυτή έχει ιδιαίτερη σημασία αφού είναι η πιο συχνά χρησιμοποιούμενη εντολή. Η εντολή SELECT έχει το παρακάτω συντακτικό:

```
SELECT name,surname FROM users ORDER BY NAME ASC;

SELECT * FROM users WHERE email='cse43833@uniwa.gr';
```

(Πηγή : <https://www.w3schools.com>)

Όπως βλέπουμε η εντολή μας δίνει την ευχέρεια να εμφανίζουμε όποια δεδομένα θέλουμε(SELECT) προσεγγίζοντας τον πίνακα κατά στήλες(WHERE) και έχοντας παράλληλα την ευκαιρία να μπορούμε να ταξινομήσουμε(ORDER BY) τα δεδομένα μας κατά την εμφάνισή τους όπως εμείς το επιθυμούμε.

Back-up Δεδομένων

Η εντολή BACKUP DATABASE χρησιμοποιείται στον SQL Server για να δημιουργηθεί ένα πλήρες αντίγραφο ασφαλείας μιας υπάρχουσας βάσης δεδομένων SQL. Η εντολή BACKUP DATABASE έχει το παρακάτω συντακτικό:

```
BACKUP DATABASE athens TO DISK = 'C:\backups\athensDB.bak';
```

(Πηγή : <https://www.w3schools.com>)

Συναθροιστηκές Συναρτήσεις

Η SQL υποστηρίζει τις ακόλουθες συναρτήσεις :

- **MIN.** Για την εύρεση της ελάχιστης τιμής μίας στήλης του πίνακα.
- **MAX.** Για την εύρεση της μέγιστης τιμής μίας στήλης του πίνακα.
- **AVG.** Για τον υπολογισμό της μέσης τιμής μίας στήλης
- **SUM.** Για τον υπολογισμό του αθροίσματος τιμών μίας στήλης του πίνακα
- **COUNT.** Για την μέτρηση των αριθμού των γραμμών

Σχεσιακό Μοντέλο και Οντότητες

Το μοντέλο οντοτήτων – συσχετίσεων βασίζεται στην θεώρηση ότι ο πραγματικός κόσμος αποτελείται από οντότητες με χαρακτηριστικά, ενώ μεταξύ των οντοτήτων διακρίνουμε συσχετίσεις. Αναπτύχθηκε ώστε κατά το σχεδιασμό μιας βάσης δεδομένων να διευκολύνεται ο ορισμός ενός σχήματος που να αναπαριστά τη συνολική δομή της βάσης. Ο συνηθέστερος τρόπος περιγραφής του μοντέλου αυτού είναι με την χρήση διαγραμμάτων οντοτήτων-συσχετίσεων ή αλλιώς ER Diagrams. Τα διαγράμματα αυτά περιέχουν έναν απλό και παράλληλα κατανοητό τρόπο περιγραφής της δομής των δεδομένων καθώς και των οντοτήτων , των συσχετίσεων και των χαρακτηριστικών της.

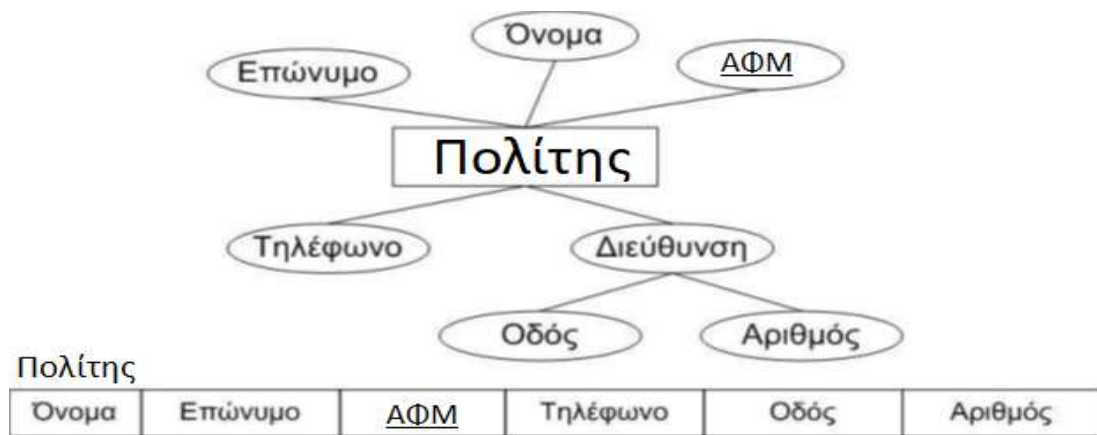
Βασικά σχήματα για ένα διάγραμμα είναι τα παρακάτω :

- ❖ **Ορθογώνιο.** Αναπαριστά ένα σύνολο οντοτήτων
- ❖ **Ρόμβος.** Αναπαριστά ένα χαρακτηριστικό μιας οντότητας
- ❖ **Έλλειψη.** Αναπαριστά ένα σύνολο συσχετίσεων

- ❖ **Ευθύγραμμο Τμήμα.** Χρησιμοποιούνται για να δηλώσουν τα χαρακτηριστικά ενός συνόλου οντοτήτων καθώς και για να δηλώσουν τη σύνδεση ενός συνόλου οντοτήτων με ένα σύνολο συσχετίσεων.

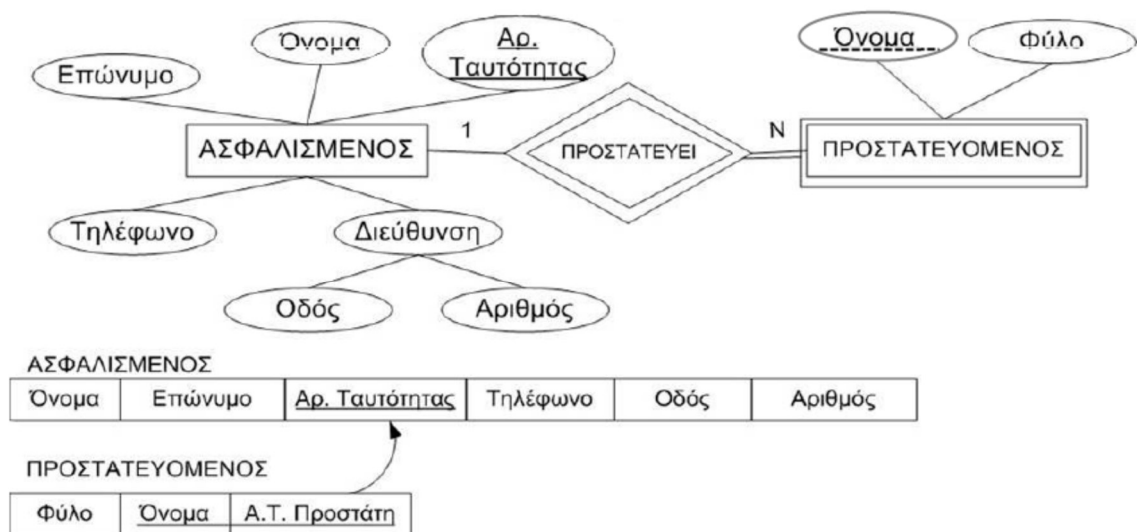
Ισχυρές Οντότητες

Για κάθε ισχυρή οντότητα δημιουργούμε μια σχέση (πίνακα) με όλα τα απλά γνωρίσματα της οντότητας. Για τα σύνθετα αποθηκεύουμε τα απλά συστατικά. Επιλέγουμε το πρωτεύον κλειδί και το υπογραμμίζουμε. Αν είναι σύνθετο, υπογραμμίζονται όλα τα απλά γνωρίσματα :



Ασθενείς Οντότητες

Για κάθε ασθενή οντότητα δημιουργούμε μια σχέση (πίνακα) στον οποίο αποθηκεύουμε το πρωτεύον κλειδί της προσδιορίζουσας οντότητας-ιδιοκτήτη. Το πρωτεύον κλειδί της οντότητας είναι συνδυασμός του πρωτεύοντος κλειδιού του ιδιοκτήτη και του μερικού κλειδιού αν υπάρχει.



Περιορισμοί Ακεραιότητας

Η ακεραιότητα αναφέρεται στη διατήρηση των δεδομένων ενός πληροφοριακού συστήματος σε μια γνωστή κατάσταση χωρίς ανεπιθύμητες τροποποιήσεις, αφαιρέσεις ή προσθήκες από μη εξουσιοδοτημένα άτομα. Παράλληλα πρέπει να δίνεται και ιδιαίτερη έμφαση σε εφεδρικά αντίγραφα της βάσης δεδομένων καθώς και η επιθυμητή ύπαρξη γρήγορων διαδικασιών ανάκαμψης σε περίπτωση απώλειας αυτών. Γενικότερα για την ακεραιότητα των δεδομένων ακολουθούμε τους παρακάτω κανονισμούς :

Περιορισμός Πεδίου Ορισμού: Η τιμή κάθε γνωρίσματος Α πρέπει να είναι μία ατομική τιμή από το πεδίο ορισμού αυτού του γνωρίσματος

AM	Όνομα	Επώνυμο	Πατρώνυμο
13456	Δημήτρης	Μαγαλιός	Νικόλαος
13457	Ανδρέας	Πλεξίδης	Βασίλειος
		Γιαννακόπουλος	
13458	Βραχνάκη	Ιωάννα	Μανώλης

Μη επιτρεπτό: Η τιμή δεν είναι ατομική

Περιορισμός Κλειδιού:

- ❖ όλες οι πλειάδες σε μία σχέση πρέπει να είναι διαφορετικές
- ❖ Τα γνωρίσματα ή συνδυασμοί των υπογήφγιων κλειδιών πρέπει να είναι μοναδικοί σε κάθε πλειάδα

Περιορισμός Ακεραιότητας Οντοτήτων

- ❖ Το πρωτεύον κλειδί δεν μπορεί να είναι NULL.

AM	Όνομα	Επώνυμο	Πατρώνυμο
13456	Δημήτρης	Μαγαλιός	Νικόλαος
13457	Ανδρέας	Πλεξίδης	Βασίλειος
		Γιαννακόπουλος	
13458	Βραχνάκη	Ιωάννα	Μανώλης

Μοναδικό AM για κάθε σπουδαστή, επομένως μπορεί να επιλεγεί ως πρωτεύον κλειδί

(Πηγή: Εργαστήριο Βάσεων Δεδομένων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Πατρών)

Σύνδεση Πινάκων

Όταν πρέπει να επιλέξουμε δεδομένα από δύο πίνακες σε μια βάση δεδομένων για να δημιουργήσουμε ένα πιο σύνθετο αποτέλεσμα, θα πρέπει να κάνουμε μια συνένωση / σύνδεση (join) μεταξύ τους. Μια SQL JOIN συνένωση χρησιμοποιείται

για να συνδυάσει σειρές από δύο ή περισσότερους πίνακες, με βάση ΕΝΑ ΚΟΙΝΟ ΠΕΔΙΟ μεταξύ τους. Μερικές από τις εντολές που μας παρέχει η MySQL για την διαχείριση των πινάκων, είναι οι παρακάτω :

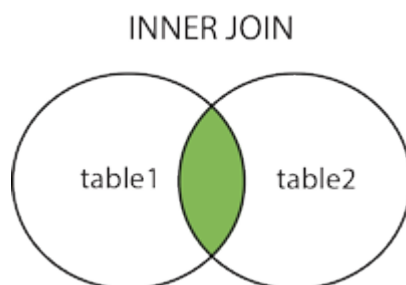
Σημείωση: Στην σύνδεση των πινάκων, ο πρώτος πίνακας που δηλώνεται πριν το JOIN είναι **πάντα ο αριστερός πίνακας**, ενώ ο πίνακας που ακολουθεί **μετά είναι πάντα ο δεξιός**.

INNER JOIN

Σύμφωνα με την σχεσιακή άλγεβρα ,η εσωτερική συνένωση (inner join) είναι η σύνδεση δυο πινάκων που στην κοινή τους στήλη , έχουν την ίδια τιμή. Όσες εγγραφές δεν έχουν κοινή τιμή και στους δυο πίνακες , δεν προβάλλονται στο τελικό αποτέλεσμα της συνένωσης.

Παράδειγμα :

```
SELECT *
FROM table1 INNER JOIN table2
ON table1.column_name =table2.column_name
```



Πίνακας Orders

Order_ID	Customer_ID	Order_Date
123	10	10-09-2019
225	20	20-12-2018
313	30	14-02-2017

Πίνακας Customers

Customer_ID	Name	Address	City
10	Tasos	Ipirou 25	Greece
20	Dimitris	Solonos 102	Greece
30	Antreas	Mesollogiou 14	Greece

40	Ioanna	Ifestou 34	Greece
50	Spiros	Vitinis 24	Greece

```
Select Orders.Order_ID, Customers.Name, Customers.Address,Orders.Order_Date
From Orders INNER JOIN Customers
ON Orders. Customer_ID = Customers. Customer_ID
```

Αποτέλεσμα θα είναι ο παρακάτω πίνακας :

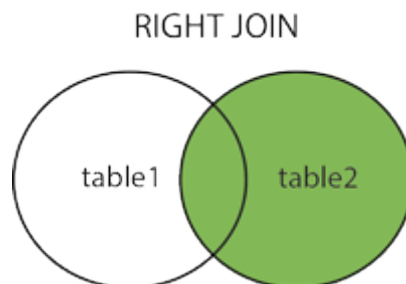
Order_ID	Name	Address	Order_Date
1	Tasos	Ipirou 25	10-09-2019
2	Dimitris	Solonos 102	20-12-2018
3	Antreas	Mesollogiou 14	14-02-2017

RIGHT JOIN

Στην δεξιά συνένωση , μας παρουσιάζονται όλα τα στοιχεία του δεξιού πίνακα , και τα κοινά στοιχεία των δυο πινάκων. Ο δεξιός πίνακας προβάλλεται ολόκληρος και από τον αριστερό πίνακα προβάλλονται μόνο τα κοινά τους σημεία.

Παράδειγμα :

```
SELECT *
FROM table1 RIGHT JOIN table2
ON table1.column_name = table2.column_name
```



Παράδειγμα :

```

SELECT Customers.Name,          Orders.Order_Date,          Customers.Address,
Customer.Country
FROM Orders RIGHT JOIN Customers
ON Orders.Customer_ID = Customers.Customer_ID

```

Αποτέλεσμα θα είναι ο παρακάτω πίνακας :

Name	Order_Date	Address	Country
Tasos	10-09-2019	Ipirou 25	Greece
Dimitris	20-12-2018	Solonos 102	Greece
Antreas	14-02-2017	Mesollogiou 14	Greece
Ioanna	NULL	Ifestou 34	Greece
Spiros	NULL	Vitinis 24	Greece

FULL JOIN

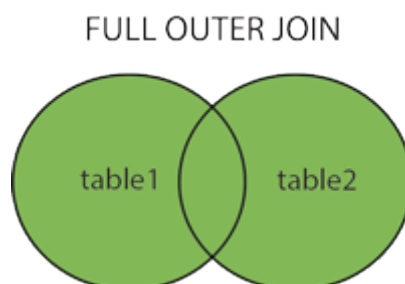
Η Full Join ή αλλιώς η πλήρης σύνδεση των δυο πινάκων, μας επιστρέφει όλα τα δεδομένα των δυο πινάκων

Παράδειγμα :

```

SELECT *
FROM table1 FULL OUTER JOIN table2
ON table1.column_name = table2.column_name;

```



Παράδειγμα :

```

SELECT Customers. Customer_ID, Customers.Name, Customers.Address,
Customers.Country, Orders. Customer_ID Orders. Order_Date
FROM Customers FULL OUTER JOIN Orders
ON Order.Customer_ID=Customers.Customer_ID

```

Αποτέλεσμα θα είναι ο παρακάτω πίνακας :

Customer_ID	Name	Address	Country	Customer_ID	Order_Date
10	Tasos	Ipirou 25	Greece	10	10-09-2019
20	Dimitris	Solonos 102	Greece	20	20-12-2018
30	Antreas	Mesollogiou 14	Greece	30	14-02-2017
40	Ioanna	Ifestou 34	Greece	Null	Null
50	Spiros	Vitinis 24	Greece	Null	Null

(Πηγή εικόνων : www.w3school.com)

ΚΕΦΑΛΑΙΟ 2 – ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΙΣΤΟΣΕΛΙΔΩΝ

Εισαγωγή

Προγραμματισμός ηλεκτρονικών υπολογιστών ονομάζεται το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος. Αυτό προκύπτει μέσα από την δημιουργία ενός συνόλου αλγορίθμων και ύστερα από προσεκτική σχεδίαση, για την αυτοματοποιημένη εκτέλεση εργασιών ή επίλυση κάποιου υπολογιστικού προβλήματος από έναν υπολογιστή. Ο προγραμματισμός περιλαμβάνει επίσης και τον έλεγχο της ορθής εκτέλεσης του προγράμματος και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εντολές που καθορίζονται στις προδιαγραφές του προγράμματος.

Η δημιουργία ενός προγράμματος σε ηλεκτρονικό υπολογιστή βασίζεται σε μια από τις χιλιάδες διαφορετικές γλώσσες προγραμματισμού δηλαδή τις προτυποποιημένες τυπικές γλώσσες απαραίτητες για τη σύνθεση ενός προγράμματος. Ανάλογα κάθε φορά την εφαρμογή υπάρχει και η εκάστοτε γλώσσα προγραμματισμού που είναι εξειδικευμένη στο αντικείμενο που θέλουμε να ασχοληθούμε και στο πρόγραμμα που θέλουμε να υλοποιήσουμε. Ο πηγαίος κώδικας του προγράμματος, το κείμενο που έχει συνταχθεί σε μία γλώσσα προγραμματισμού, πρέπει στη συνέχεια να μεταφραστεί σε γλώσσα μηχανής από εξειδικευμένο λογισμικό το οποίο ονομάζεται μεταγλωττιστής ή αλλιώς compiler. Το πρόγραμμα δηλαδή στο οποίο συντάσσεις τον κώδικα στην εκάστοτε γλώσσα, σου το αποσφαλματώνει και τελικά σου δημιουργεί το εκτελέσιμο προϊόν ώστε να είναι σε θέση να εκτελεστεί από τον υπολογιστή.

Γλώσσα προγραμματισμού λέγεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Κάθε γλώσσα προγραμματισμού (όπως άλλωστε και οι ανθρώπινες γλώσσες) ορίζεται με το δικό της συντακτικό, και τους δικούς της λεκτικούς κανόνες, που ορίζουν τη δομή και το νόημα αντίστοιχα, των προτάσεων της γλώσσας. Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών, αλλά και για την ακριβή διατύπωση αλγορίθμων. Ορισμένοι ειδικοί χρησιμοποιούν τον όρο γλώσσα προγραμματισμού μόνο για τυπικές γλώσσες που μπορούν να εκφράσουν όλους τους πιθανούς αλγορίθμους. Μη-υπολογιστικές γλώσσες όπως η HTML δεν λέγονται συνήθως γλώσσες προγραμματισμού.

Υπάρχουν χιλιάδες διαφορετικές γλώσσες προγραμματισμού, και κάθε χρόνο δημιουργούνται περισσότερες. Χαρακτηριστικό παράδειγμα είναι η γλώσσα

προγραμματισμού C η οποία πρωτοεμφανίστηκε το 1972 από τον Dennis Ritchie. Το 1983 ο Bjarne Stroustrup βασιζόμενος στην C , δημιούργησε την C++ .Πάνω στην τελευταία έχουν βασιστεί η C# (2000) αλλά και η παγκοσμίως διαδεδομένη πλέον γλώσσα προγραμματισμού Java(1995).

(Πηγή : Ο Προγραμματισμός – www.wikipedia.org. Et.2016)

PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Αυτό σημαίνει ότι ένα site αποκτά διαδραστικότητα π.χ: η εισαγωγή ενός username και ενός κωδικού σε μια σελίδα και η επαλήθευσή τους ώστε ο χρήστης να συνδεθεί σ' αυτό το site

Ένα αρχείο PHP μπορεί να περιέχει κείμενο, κώδικα HTML, CSS, JavaScript και PHP, με τον κώδικα να εκτελείται στον server και το αποτέλεσμα να επιστρέφει στον browser ως απλές σελίδες HTML.

Αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Η ευρύτητα στη χρήση της είναι απόρροια της ευκολίας που παρουσιάζει ο προγραμματισμός με αυτή αλλά και στο γεγονός πως είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή. Είναι μια γλώσσα προγραμματισμού που διανέμεται ελεύθερα στους προγραμματιστές. Για την συντήρηση και την ανάπτυξη της PHP σε οποιαδήποτε νέα ανάγκη παρουσιαστεί , υπάρχει μια ομάδα με προγραμματιστές οι οποίοι με την κατάλληλη τεχνογνωσία αναλαμβάνουν την συντήρηση της γλώσσας και την αναβάθμιση αυτής. Τέτοιες ανάγκες μπορεί να είναι, είτε ασφάλειας είτε διόρθωση μιας προηγούμενης έκδοσης λόγω κάποιου σφάλματος ή δυσλειτουργίας που υπάρχει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

Η PHP μπορεί να δημιουργήσει, να ανοίξει, να κλείσει, να διαβάσει, να γράψει και να διαγράψει αρχεία σε έναν server. Μπορεί να στείλει και να λάβει cookies. Μπορεί επίσης να προσθέσει, να διαγράψει και να τροποποιήσει δεδομένα στη Βάση Δεδομένων σας. Ακόμη μπορεί να κρυπτογραφήσει δεδομένα. Με την PHP δεν περιορίζεσαι στο να παράγεις HTML. Μπορείς να παράγεις εικόνες, αρχεία PDF, ακόμα και ταινίες Flash, καθώς και οποιοδήποτε κείμενο όπως XHTML και XML. Η PHP τρέχει σε όλες τις πλατφόρμες όπως Windows, Linux, Unix, Mac OS X κ.λπ. Είναι συμβατή με σχεδόν όλους τους server που χρησιμοποιούνται σήμερα (Apache, IIS, κ.λπ.) Υποστηρίζει ένα ευρύ φάσμα από Βάσεις Δεδομένων, είναι δωρεάν, είναι εύκολη να τη μάθει κανείς και τρέχει αποτελεσματικά στους server. Προγράμματα όπως το

Xampp, Wampp, Lampp και το Mamp(το πρώτο γράμμα αντιπροσωπεύει το λογισμικό πάνω στο οποίο θα εγκατασταθεί η PHP) κάνουν εφικτή την λειτουργία της σε όλα τα παραπάνω συστήματα.

Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού C[1] ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0 αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 5, 6 και 7 της PHP.

Σύνταξη της PHP

Η PHP, ως γλώσσα σεναριακού προγραμματισμού δυναμικών ιστοσελίδων, σχεδιάστηκε ώστε να ενσωματώνεται σε έγγραφα HTML με ιδιαίτερη ευκολία. Για αυτό τον σκοπό ο κώδικας πρέπει να είναι απαραίτητα εντός της ειδικής ετικέτας `<?php ...?>` ή της συντομευμένης μορφής `<?...?>` που ονομάζεται και short open tag. Η τελευταία συνήθως αποφεύγεται διότι είναι απενεργοποιημένη στο php.ini μέσω της οδηγίας `“short_open_tag=OFF”`. Αν παρόλα αυτά κάποιος επιθυμεί να κάνει χρήση της, θα πρέπει να τροποποιηθεί η οδηγία ώστε να έχει την τιμή `“short_open_tag=ON”`.

Κάθε αρχείο πηγαίου κώδικα php, με κώδικα html θα πρέπει να έχει κατά σύμβαση την κατάληξη .php, όταν αποθηκεύεται αντί για την κλασική κατάληξη .html. Αυτό θα πρέπει να γίνεται όχι μόνο στην ανάγκη διαχωρισμού των δυναμικών ιστοσελίδων από τις στατικές αλλά και για πρακτικούς λόγους που έχουν να κάνουν με την λειτουργία του λογισμικού του Web Server. Ένα script PHP μπορεί να τοποθετηθεί οπουδήποτε στο έγγραφο.

```

<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>

```

Στην PHP, οι λέξεις-κλειδιά (π.χ., if, else, while, echo κλπ.) δεν διαφέρουν από πεζά-κεφαλαία. Στο παρακάτω παράδειγμα, και οι τρεις παρακάτω δηλώσεις echo είναι αποδεκτές:

```

<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>

```

Αυτό όμως δεν συμβαίνει με τις μεταβλητές '\$variable'

Μεταβλητές στην PHP

Μεταβλητή ορίζεται μια συμβολική ονομασία μια τιμής η οποία μπορεί να αλλάξει τιμές κατά το πέρασμα του χρόνου.

Κανονισμός μεταβλητών της PHP:

- ❖ Ξεκινούν με \$.
- ❖ Ακολουθεί γράμμα ή underscore (_).
- ❖ Μπορεί να περιέχει γράμματα, αριθμούς, underscores (_) ή dashes (\, /)
- ❖ Δεν μπορεί να περιέχει κενά.
- ❖ Είναι case sensitive (\$onoma και \$ONOMA αποτελούν δυο ξεχωριστές μεταβλητές)

Παραδείγματα:

```
<?php
  $txt= "Hello world!"
  $x= 5;
  $X= 11;
?>
```

Έτσι εν τέλει μετά την εκτέλεση των εντολών η μεταβλητή \$txt θα έχει την τιμή Hello world! ,η τιμή x θα έχει την τιμή 5 και η μεταβλητή X θα έχει την τιμή 11.

Υπερκαθολικές Μεταβλητές

Ορισμένες προκαθορισμένες μεταβλητές στην PHP ορίζονται ως "superglobals» ή διαφορετικά «υπερκαθολικές». Αυτό σημαίνει ότι είναι πάντα προσβάσιμες , ανεξάρτητα από το πεδίο εφαρμογής. Επίσης βασικό πλεονέκτημα των υπερκαθολικών μεταβλητών είναι η πρόσβαση από οποιαδήποτε λειτουργία, την τάξη ή το αρχείο χωρίς να χρειάζεται να κάνουμε κάτι το ιδιαίτερο.

Οι υπερκαθολικές μεταβλητές της PHP είναι:

- ❖ \$_GLOBALS
- ❖ \$_SERVER
- ❖ \$_REQUEST
- ❖ \$_POST
- ❖ \$_GET
- ❖ \$_FILES
- ❖ \$_ENV
- ❖ \$_COOKIE
- ❖ \$_SESSION

Οι πίνακες \$_GET και \$_POST

Είναι οι πιο σημαντικοί πίνακες τις κατηγορίας τους. Τα περιεχόμενα τους, τα οποία συμπληρώνονται αυτόματα από το διερμηνευτή της PHP, ανταποκρίνονται στα δεδομένα που αποστέλλει ο Web Browser προς τον Web Server μέσω HTTP αιτήσεων GET και POST αντίστοιχα. Με την χρήση των πινάκων αυτών κατά περίπτωση , τα σενάρια της εφαρμογής επιτρέπουν την αλληλεπίδραση με το χρήστη και την παραμετροποιημένη εμφάνιση των δυναμικών ιστοσελίδων της εφαρμογής. Η κύρια διαφορά μεταξύ \$_GET και \$_POST είναι πως η πρώτη είναι μία παράταξη μεταβλητών η οποία μεταφέρεται στο επόμενο script μέσω των παραμέτρων του URL , ενώ το \$_POST μεταφέρει τα δεδομένα από το αίτημα της HTTP POST.

Οι πληροφορίες που αποστέλλονται από μια φόρμα με τη μέθοδο GET είναι ορατές σε όλους (όλα τα ονόματα των μεταβλητών καθώς και οι τιμές που περιέχουν εμφανίζονται στο URL).Το GET έχει επίσης όρια για το ποσό των πληροφοριών για την αποστολή . Ο περιορισμός είναι περίπου 2000 χαρακτήρες . Ωστόσο , επειδή οι

μεταβλητές εμφανίζονται στη διεύθυνση URL, είναι δυνατόν να κρατήσουμε σελιδοδείκτη της σελίδας στον web browser . Αυτό μπορεί να είναι χρήσιμο σε ορισμένες περιπτώσεις.

Αντίθετα, με το POST οι πληροφορίες που αποστέλλονται από μια φόρμα είναι αόρατες στους άλλους (Ονόματα μεταβλητών / Τιμές μεταβλητών) ενσωματωμένα μέσα στο σώμα του αιτήματος HTTP POST) και δεν έχει όρια στην ποσότητα των πληροφοριών για την αποστολή .Αυτός είναι ο κυριότερος λόγος που οι προγραμματιστές προτιμούν την χρήση του \$_POST από το \$_GET.

Σημείωση : Η μέθοδος GET δεν πρέπει ΠΟΤΕ να χρησιμοποιηθεί για την αποστολή κωδικών πρόσβασης ή άλλων ευαίσθητων πληροφοριών !

Ο πίνακας \$_SESSION

Τα sessions ή συνεδρίες είναι ένας μηχανισμός με τον οποίο μπορούμε να ελέγξουμε την ταυτότητα του επισκέπτη όταν αυτός περιηγείται στις σελίδες ενός ιστότοπου. Σκοπό έχει ,να επιτρέψει σε μια εφαρμογή να διαχειρίζεται και να αποθηκεύει πληροφορίες του χρήστη στον ίδιο τον Web Server,την τοποθεσία η οποία θεωρείται περισσότερο ασφαλής από τον υπολογιστή. Με την php μπορούμε να δημιουργήσουμε sessions για τους επισκέπτες και έτσι να ελέγχετε την ταυτότητά τους και κατά συνέπεια την πρόσβαση αυτών σε ευαίσθητες σελίδες. Τα session χρησιμοποιούν την τεχνολογία των cookies. Δηλαδή κάθε επισκέπτης δέχεται ένα cookie στο μηχανήμα του το οποίο περιέχει έναν αριθμό (id) το οποίο είναι και το αναγνωριστικό του. Βασική προϋπόθεση είναι η χρήση της συνάρτησης session_start(),για την δημιουργία μιας νέας συνεδρίας ή για την φόρτωση μιας υπάρχουσας συνεδρίας.

Η Ψευδοσυνάρτηση echo

Η echo δεν είναι μια πραγματική συνάρτηση της PHP, άλλα μια φραστική δομή. Η οποία δέχεται ως όρισμα μια έκφραση δεδομένων (expression).Η έκφραση αυτή αξιολογείται, μετατρέπεται σε παράσταση κειμένου και μεταφέρεται στην έξοδο. Συνηθέστερα , μια έκφραση αποτελείται από κάποιο αριθμό μεταβλητών βαθμωτού τύπου ή/και σταθερών συμβολοσειρών. Η echo δεν μπορεί να λάβει ως μέρος της έκφρασης έναν ολόκληρο πίνακα, μπορεί όμως αν χρησιμοποιηθεί για να εμφανίσει συγκεκριμένα στοιχεία του πίνακα , αρκεί αυτά να είναι βαθμωτού τύπου .

Ακολουθούν παραδείγματα που παρουσιάζουν την χρήση της echo για διάφορους τύπους δεδομένων:

```
$a=2016;  
$pi=3.14;  
$m=2  
$sonoma=Panagiotis
```

```
Εκτελούμε την echo:  
Echo $a;  
Echo $pi;  
Echo $m;  
Echo $sonoma
```

Στην έξοδο του σεναρίου θα εμφανιστούν τα ακόλουθα :

```
2016  
3,14  
2  
Panagiotis
```

Όπως βλέπουμε της λειτουργίας ελεγχόμενης εξόδου πληροφοριών που παρέχει η echo καθίσταται ως η πιο χρήσιμη δήλωση της PHP.

Πηγές : Ανάπτυξη Διαδουκτουακών εφαρμογών – Παναγιώτης Δ. Κεντερλής ετ.2016

www.w3schools.com ετ.2016

www.php.net ετ.2016

Βασικές Εντολές PHP ,Αντώνης Χρυσόπουλος ,Κατερίνα Κυπριώτη, Στέλιος Μόσχογλου Θεοδόσης Σουργκούνης

JavaScript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript

Η Ιστορία της Javascript

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. Η LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφάρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην `java.util.Date` τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (μαζί με άλλους λόγους). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαϊσίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side)

Μοντέλο Εκτέλεσης

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

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

ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

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

JQUERY

Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα. Η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT.

Χαρακτηριστικά

- ❖ DOM element επιλογές χρησιμοποιώντας την ανοιχτού κώδικα μηχανή επιλογής πολλαπλών φυλλομετρητών Sizzle.
- ❖ DOM διάσχιση και τροποποίηση (υποστηρίζοντας CSS 1-3)
- ❖ χειρισμός DOM βασισμένος σε CSS επιλογείς που χρησιμοποιεί τα id και class σαν κριτήρια για να κατασκευάσει επιλογείς.
- ❖ Events
- ❖ Εφέ και κινητά στοιχεία
- ❖ AJAX
- ❖ Επεκτασιμότητα μέσω plug-ins
- ❖ Εργαλεία όπως πληροφορίες user-agent, ανίχνευση χαρακτηριστικών.
- ❖ Μεθόδους συμβατότητας που είναι εγγενώς διαθέσιμα σε σύγχρονα προγράμματα περιήγησης.
- ❖ Υποστήριξη πολλαπλών φυλλομετρητών.

Χρήση της βιβλιοθήκης

Η jQuery είναι ένα αρχείο JavaScript, που περιέχει όλες τις λειτουργίες. Μπορεί να συμπεριληφθεί σε μια ιστοσελίδα παρέχοντας το αρχείο τοπικά

```
<script type="text/javascript" src="jquery.js"></script>
```

ή έχοντας έναν σύνδεσμο σε έναν από τους πολλούς διακομιστές που τη φιλοξενούν.

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">
</script>
```

ΚΕΦΑΛΑΙΟ 3 – ΣΧΕΔΙΑΣΜΟΣ ΙΣΤΟΣΕΛΙΔΩΝ

Σχεδιασμός Σελίδων Ιστού (Web Daring)

Ο σχεδιασμός στο Διαδίκτυο αφορά την ικανότητα δημιουργίας παρουσιάσεων περιεχομένου (συνήθως κειμένου ή πολυμέσων) οι οποίες φτάνουν στον τελικό-χρήστη μέσω του Παγκόσμιου Ιστού, με τη χρήση λογισμικού όπως ένας φυλλομετρητής (web browser) ή άλλου λογισμικού σχεδιασμένου για το διαδίκτυο όπως η τηλεόραση μέσω διαδικτύου, κινητών τηλεφώνων κλπ.

Η ιστοσελίδα είναι ένα ηλεκτρονικό αρχείο ή ένα σύνολο από ηλεκτρονικά αρχεία που υπάρχουν σε έναν ή και περισσότερους εξυπηρετητές (server/servers) και παρουσιάζει κείμενα και εφαρμογές πολυμέσων στον τελικό-χρήστη. Τέτοια στοιχεία όπως κείμενο, εικόνες (σύμφωνα με τα πρότυπα SVG, BMP, GIF, JPEG ή PNG) και φόρμες μπορούν να τοποθετηθούν στη σελίδα με τη χρήση γλώσσων σήμανσης υπερκειμένου όπως HTML/XHTML/XML. Η αναπαραγωγή πιο σύνθετων πολυμέσων (ανυσματικών γραφικών, βίντεο, ήχων, γραφικών με ενσωματωμένο ήχο και εικόνα) μπορεί να γίνει με πρόσθετα (plug-ins) όπως το Flash, το QuickTime, το περιβάλλον χρόνου εκτέλεσης Java, κ.α ή με τεχνολογίες όπως οι γλώσσες σήμανσης (X) HTML5, XML και MathML, και το πρότυπο φύλλων στυλ CSS 3.

Οι καινούριες εκδόσεις των προγραμμάτων περιήγησης (Internet Explorer 7, 8, Firefox 3.6, safari, chromium κλπ) που ακολουθούν τα W3C πρότυπα οδήγησαν σε μια ευρεία αποδοχή και χρήση των XHTML/XML σε συνδυασμό με τα CSS (Cascading Style Sheets) για την τοποθέτηση και διαχείριση των στοιχείων και αντικειμένων της ιστοσελίδας. Τα τελευταία πρότυπα στοχεύουν στο να αποκτήσουν τα προγράμματα περιήγησης την δυνατότητα να προσφέρουν μια ευρεία γκάμα επιλογής πολυμέσων και πρόσβασης στους πελάτες χωρίς τη χρήση των προσθέτων (plug-ins).

Γενικότερα οι ιστοσελίδες διαχωρίζονται σε στατικές και δυναμικές:

- ❖ Οι στατικές δεν αλλάζουν περιεχόμενο και διάταξη (layout) με οποιοδήποτε αίτημα εκτός και αν ο προγραμματιστής αναβαθμίσει (update) τη σελίδα. Μια απλή HTML σελίδα είναι παράδειγμα στατικού περιεχομένου.

- ❖ Οι δυναμικές προσαρμόζουν το περιεχόμενο και/ή την εμφάνισή τους σύμφωνα με την καταχώρηση/αλληλεπίδραση ή τις αλλαγές του τελικού χρήστη στο περιβάλλον προγραμματισμού (χρήστης, ώρα, τροποποιήσεις στη βάση δεδομένων, κτλ) Το περιεχόμενο μπορεί να αλλάζει στον υπολογιστή του τελικού-χρήστη με τη χρήση των γλωσσών προγραμματισμού που εκτελούνται στον υπολογιστή του χρήστη (JavaScript, VBScript, Actionscript, etc.). Το περιεχόμενο στις δυναμικές σελίδες συχνά μεταφράζεται στον εξυπηρετητή (server) μέσω γλωσσών προγραμματισμού που εκτελούνται στον εξυπηρετητή (Perl, PHP, ASP, JSP, ColdFusion, .NET κτλ).

Με την συνεχόμενη ειδίκευση στην τεχνολογία της Πληροφορικής δημιουργείται η ανάγκη διαχωρισμού του Σχεδιασμού Ιστοσελίδων από τον Προγραμματισμό Ιστοσελίδων.

Για τη διαδικασία σχεδιασμού μιας ιστοσελίδας, μιας εφαρμογής ή ενός πολυμέσου για το διαδίκτυο μπορεί να συνδυάζονται πολλοί κλάδοι όπως animation, συγγραφή, επικοινωνιακός σχεδιασμός, εταιρική ταυτότητα, σχεδιασμός γραφικών, αλληλεπίδραση ανθρώπου-υπολογιστή, αρχιτεκτονική υπολογιστών, σχεδιασμός αλληλεπίδρασης, αγοραλογία, φωτογραφία, βελτιστοποίηση μηχανών αναζήτησης και τυπογραφία.

Τα εργαλεία και οι τεχνολογίες που χρησιμοποιούνται περιλαμβάνουν μεταξύ άλλων:

- ❖ Γλώσσες σήμανσης: HTML, XHTML, XML
- ❖ Γλώσσες φύλλων ύφους: CSS, XML
- ❖ Σενάρια στον πελάτη (Client-side scripting): JavaScript
- ❖ Σενάρια στον εξυπηρετητή (Server-side scripting): PHP, ASP, Perl, Ruby/Ruby on Rails
- ❖ Τεχνολογίες βάσεων δεδομένων: MySQL, PostgreSQL, Microsoft SQL Server
- ❖ Τεχνολογίες πολυμέσων: Flash, Silverlight

Οι ιστοσελίδες μπορούν να είναι στατικές ή δυναμικές. που αναπροσαρμόζουν αυτόματα το περιεχόμενό τους ή την εμφάνισή τους βασιζόμενες σε μια ποικιλία παραγόντων, όπως μια καταχώρηση από τον τελικό-χρήστη, μια καταχώρηση ή αλλαγή στο περιβάλλον προγραμματισμού από τον χειριστή της ιστοσελίδας (όπως εν παραδείγματι μια μορφοποίηση της βάσης δεδομένων).

Εξαιτίας της συνεχούς εξειδίκευσης στους τομείς της επικοινωνιακής Πληροφορικής, δημιουργείται μια σαφής τάση διαχωρισμού του σχεδιασμού στο Διαδίκτυο από την ανάπτυξη του διαδικτύου ως μέσου ροής της πληροφορίας και των αγαθών προς όλες τις διαδικτυακές υπηρεσίες.

Προσβασιμότητα Ιστοσελίδων

Για να είναι προσβάσιμη μια ιστοσελίδα θα πρέπει να ακολουθεί κάποιες αρχές προσβασιμότητας. Αυτές οι αρχές είναι γνωστές ως WCAG όταν μιλάμε για το περιεχόμενο και μπορούν να ομαδοποιηθούν στις ακόλουθες κατηγορίες.

- ❖ Χρήση σημασιολογικής σήμανσης (semantic markup), η οποία προσφέρει μια ενιαία δομή στο αρχείο.
- ❖ Το semantic markup επίσης αναφέρεται στην οργάνωση της δομής της ιστοσελίδας αλλά και στην ανακοίνωση/έκδοση/δημιουργία ορισμών των διαδικτυακών υπηρεσιών με τρόπο ώστε να είναι αναγνωρίσιμες από άλλες διαδικτυακές υπηρεσίες σε διαφορετικές ιστοσελίδες. Τα πρότυπα για το semantic web θέτονται βάσει IEEE.
- ❖ Χρήση έγκυρων γλωσσών σήμανσης που να ανταποκρίνονται σε ένα δημοσιευμένο DTD ή Schema.
- ❖ Να παρέχει το ισοδύναμο της πληροφορίας σε κείμενο για κάθε πληροφορία που παρέχεται με γραφικά ή πολυμέσα.
- ❖ Να χρησιμοποιεί συνδέσμους ενσωματωμένους στο κείμενο.
- ❖ Να μην χρησιμοποιεί πλαίσια.
- ❖ Χρήση CSS αντί για HTML για την κατασκευή πινάκων.
- ❖ Συγγραφή της σελίδας με τρόπο τέτοιο ώστε όταν ο κώδικας διαβαστεί γραμμή προς γραμμή από user agents (όπως είναι τα screen readers) να παραμένει ευανάγνωστος.

Παρόλα αυτά η W3C επιτρέπει ως εξαίρεση στους πίνακες που προορίζονται για εμφάνιση στην ιστοσελίδα να παραμένουν κατανοητοί όταν γραμμικοποιούνται ή σε περίπτωση που κάτι αντίστοιχο μπορεί να εφαρμοστεί.

Η προσβασιμότητα των ιστοσελίδων αλλάζει επίσης εξαιτίας των Συστημάτων Διαχείρισης Περιεχομένου (Content Management Systems), τα οποία επιτρέπουν να γίνονται αλλαγές στις ιστοσελίδες χωρίς να απαιτείται ιδιαίτερη γνώση από προγραμματισμό.

Η Ανάπτυξη ιστοσελίδων απαιτεί τη συνεργασία πολλών διαφορετικών στοιχείων ώστε το διαδίκτυο να είναι προσβάσιμο σε ανθρώπους με αναπηρίες. Αυτά τα στοιχεία περιλαμβάνουν:

- ❖ Περιεχόμενο – η πληροφορία που υπάρχει σε μια ιστοσελίδα ή σε κάποια εφαρμογή του διαδικτύου περιλαμβάνει:
 - Τη φυσική πληροφορία όπως κείμενο, εικόνες και ήχους
 - Κώδικα ή επισήμανση που διευκρινίζει τη δομή, παρουσίαση κλπ.
- ❖ Προγράμματα πλοήγησης, προγράμματα αναπαραγωγής πολυμέσων κλπ.
- ❖ Υποβοηθητική τεχνολογία, για ορισμένες περιπτώσεις – screen readers, εναλλακτικό πληκτρολόγιο, διακόπτες, προγράμματα σάρωσης κλπ.
- ❖ Γνώση του επιπέδου των χρηστών, των εμπειριών τους και σε ορισμένες περιπτώσεις προσαρμοσμένες στρατηγικές με χρήση του διαδικτύου

- ❖ Προγραμματιστές-Σχεδιαστές, συγγραφείς κτλ, συμπεριλαμβανομένου Προγραμματιστών με αναπηρίες και χρηστών που συνεισφέρουν στη διαμόρφωση του περιεχομένου
- ❖ Συντακτικών εργαλείων – προγραμμάτων που κατασκευάζουν ιστοσελίδες
- ❖ Εργαλεία Αξιολόγησης – Εργαλείων αξιολόγησης της Διαδικτυακή Προσβασιμότητας, HTML εγγυητών (HTML validators), CSS εγγυητών (CSS validators), κλπ.

Η Γλώσσα Σήμανσης HTML

Ορισμός

Η HTML (HyperText Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML και από στατικές τις κάνουν διαδραστικές.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και

τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Ιστορία της Html

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», όπως την γενική παρουσίαση πληροφοριών (π.χ. μια εγκυκλοπαίδεια), στοχευμένη δημοσίευση (κείμενα βοήθειας, τεκμηρίωσης, εκπαίδευσης, κλπ.) μέχρι ακόμα και για καταγραφή προσωπικών σημειώσεων.

Πρώτες προδιαγραφές

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά την λήξη των πρόχειρων HTML και HTML+, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασιζόνταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης απόψεων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με την σύμφωνη γνώμη των δημιουργών του λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

Σήμανση

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), τους βασισμένους σε χαρακτήρες τύπους δεδομένων, τις αναφορές χαρακτήρων και τις αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου, η οποία ορίζει στο πρόγραμμα περιήγησης (browser) τον τρόπο εμφάνισης της σελίδας.

Στην HTML, το πρόγραμμα Hello World, ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών

σεναρίων και γλωσσών σήμανσης, φτιάχνεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στην HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

(Το κείμενο ανάμεσα στο <html> και το </html> περιγράφει την ιστοσελίδα, και το κείμενο μεταξύ του <body> και του </body> είναι το ορατό μέρος της. Το σημασμένο κείμενο <title>Hello HTML</title> καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser.)

Η δήλωση τύπου εγγράφου (το <!DOCTYPE html>) στον πιο πάνω κώδικα είναι για την HTML5. Αν δεν συμπεριλαμβάνεται η δήλωση <!doctype html>, τότε μερικοί browser θα καταφύγουν στην λειτουργία quirks για την εμφάνιση της σελίδας.

Στοιχεία

Τα έγγραφα HTML αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή < και >.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: <ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">περιεχόμενο</ετικέτα>. Μερικά στοιχεία HTML περιγράφονται ως άδεια ή κενά στοιχεία, έχουν τη μορφή <ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια πλάγια «/», η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

Παραδείγματα στοιχείων

Κεφαλίδα του εγγράφου HTML: `<head>...</head>`. Συνήθως περιέχει τον τίτλο, π.χ.:

```
<head>
  <title>Ο τίτλος</title>
</head>
```

Επικεφαλίδες: οι επικεφαλίδες στην HTML ορίζονται με τις ετικέτες `<h1>` έως `<h6>`

```
<h1>Επικεφαλίδα1</h1>
<h2>Επικεφαλίδα2</h2>
<h3>Επικεφαλίδα3</h3>
<h4>Επικεφαλίδα4</h4>
<h5>Επικεφαλίδα5</h5>
<h6>Επικεφαλίδα6</h6>
```

Παράγραφοι:

```
<p>Παράγραφος 1</p>
<p>Παράγραφος 2</p>
```

Αλλαγή γραμμής: `
`. Η διαφορά ανάμεσα στο `
` και το `<p>` είναι ότι το «`br`» αλλάζει γραμμή χωρίς να αλλάζει την σημαντική δομή της σελίδας, ενώ το «`p`» τεμαχίζει τη σελίδα σε παραγράφους. Το «`br`» είναι ένα άδειο στοιχείο, δηλαδή δεν έχει περιεχόμενο, ούτε και χρειάζεται ετικέτα τερματισμού.

```
<p>Αυτή <br> είναι μία παράγραφος <br> με <br> αλλαγές γραμμής</p>
```

σύνδεσμος σε HTML: Για τη δημιουργία του χρησιμοποιείται η ετικέτα `<a>`. Η ιδιότητα `href` περιέχει τη σελίδα προορισμού του συνδέσμου.

```
<a href="https://www.google.gr">Ένας σύνδεσμος στο Google</a>
```

Σχόλια: `<!-- Αυτό είναι ένα σχόλιο -->`

Τα σχόλια μπορούν να βοηθήσουν στην κατανόηση της σήμανσης. Δεν εμφανίζονται στην ιστοσελίδα.

Υπάρχουν διάφοροι τύποι στοιχείων στην HTML.

Η δομική σήμανση περιγράφει τον σκοπό του κειμένου

Για παράδειγμα, το `<h2>Golf</h2>` σημαίνει ότι η λέξη «Golf» θα είναι μια επικεφαλίδα δεύτερου επιπέδου. Η δομική σήμανση δεν εμπεριέχει συγκεκριμένο τρόπο εμφάνισης, αλλά οι περισσότεροι φυλλομετρητές έχουν δικό τους προκαθορισμένο τρόπο για τη μορφοποίηση όλων των στοιχείων. Το περιεχόμενο μπορεί να μορφοποιηθεί οπτικά με τη χρήση των CSS.

Η σήμανση οπτικής μορφοποίησης περιγράφει την εμφάνιση του κειμένου, *άσχετα από τον σκοπό του.*

Για παράδειγμα το `έντονο κείμενο` υποδεικνύει ότι οι συσκευές που θα εμφανίσουν το κείμενο θα το κάνουν έντονο, αλλά δεν λέει τι θα κάνουν με το κείμενο οι συσκευές που δεν μπορούν να το εμφανίσουν, όπως για παράδειγμα οι συσκευές φωνητικής ανάγνωσης σελίδων. Τόσο στην περίπτωση του `έντονου` όσο και του `<i>πλάγιου</i>`, υπάρχουν άλλα στοιχεία τα οποία μπορεί οπτικά να εμφανίζονται ίδια, αλλά είναι πιο σημαντικά, όπως το `σημαντικό κείμενο` και το `κείμενο με έμφαση` αντίστοιχα. Είναι ευκολότερο να ερμηνεύσει ένας φωνητικός φυλλομετρητής τα τελευταία δύο στοιχεία. Ωστόσο, δεν είναι ισότιμα με τα αντίστοιχα στοιχεία οπτικής μορφοποίησης. Για παράδειγμα ένα πρόγραμμα φωνητικής ανάγνωσης της σελίδας δεν θα πρέπει να προφέρει με έμφαση τον τίτλο ενός βιβλίου, αλλά όταν το ίδιο περιεχόμενο εμφανίζεται σε μια οθόνη τότε ο τίτλος θα είναι με πλάγια γράμματα. Τα περισσότερα στοιχεία οπτικής μορφοποίησης έχουν θεωρηθεί ξεπερασμένα μετά την προδιαγραφή HTML 4.0, και έχουν αντικατασταθεί από τα CSS.

Η σήμανση υπερκειμένου κάνει μερικά τμήματα ενός εγγράφου να συνδέουν με άλλα έγγραφα.

Το στοιχείο `anchor` δημιουργεί έναν υπερσύνδεσμο στο έγγραφο, και η ιδιότητα `href` ορίζει τον στόχο του συνδέσμου. Για παράδειγμα, η σήμανση HTML `Βικιπαίδεια` εμφανίζει το κείμενο «Βικιπαίδεια» ως υπερσύνδεσμο. Για την εμφάνιση μιας εικόνας ως συνδέσμου, μπορεί να εισαχθεί ένα στοιχείο `` ως περιεχόμενο του στοιχείου «`a`». Όπως και το `
`, το `` είναι άδειο στοιχείο. Έχει ιδιότητες αλλά δεν έχει περιεχόμενο, ούτε ετικέτα τερματισμού: ``.

Ιδιότητες

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης ενός στοιχείου, μετά το όνομα του στοιχείου. Η τιμή μπορεί να περικλείεται σε μονά ή διπλά εισαγωγικά, παρότι τιμές που αποτελούνται από συγκεκριμένους χαρακτήρες μπορούν να γράφονται χωρίς εισαγωγικά στην HTML, αλλά όχι στην XHTML. Το να μένουν οι τιμές των ιδιοτήτων χωρίς εισαγωγικά θεωρείται ανασφαλές. Εκτός από τις ιδιότητες που γράφονται ως ζεύγη ονομάτων και τιμών, υπάρχουν και μερικές οι οποίες επηρεάζουν το στοιχείο απλά με την παρουσία τους μέσα στην ετικέτα εκκίνησης, όπως η ιδιότητα `ismap` του στοιχείου `img`.

Υπάρχουν και μερικές κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

- ❖ Η ιδιότητα `id` παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο. Χρησιμοποιείται για να ταυτοποιεί το στοιχείο ώστε τα CSS να μπορούν να αλλάξουν τον τρόπο που αυτό εμφανίζεται, καθώς και τα σενάρια μπορούν να αλλάξουν, να μετακινήσουν ή να διαγράψουν τα περιεχόμενα ή την εμφάνισή του. Ακόμα, αν ένα `id` προστεθεί στο URL μιας σελίδας, παρέχει ένα μοναδικό αναγνωριστικό για ένα τμήμα της σελίδας, για παράδειγμα: <http://el.wikipedia.org/wiki/HTML#Ιδιότητες>
- ❖ Η ιδιότητα `class` παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης. Για παράδειγμα, ένα έγγραφο HTML μπορεί να χρησιμοποιεί την επισήμανση `class="notation"` σε μερικά στοιχεία για να ξεχωρίσει από το υπόλοιπο κείμενο του εγγράφου. Κατά την εμφάνιση του εγγράφου, αυτά τα στοιχεία μπορεί -για παράδειγμα- να εμφανίζονται όλα μαζί στο τέλος της σελίδας ως υποσημειώσεις, άσχετα με την θέση που εμφανίζονται μέσα στον κώδικα. Επίσης οι ιδιότητες `class` χρησιμοποιούνται σημασιολογικά στα `microformat`. Ένα στοιχείο μπορεί να έχει πολλαπλές κλάσεις, για παράδειγμα το `class="notation important"` βάζει το στοιχείο τόσο στην κλάση «`notation`» όσο και στην «`important`».
- ❖ Η ιδιότητα `style` εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες `id` ή `class` ώστε να επιλέγεται το στοιχείο μέσα σε ένα CSS, αλλά μερικές φορές μπορεί να είναι πιο απλό να ανατεθούν `style` κατευθείαν στο στοιχείο.
- ❖ Η ιδιότητα `title` προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Στους περισσότερους φυλλομετρητές αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας (`tooltip`).

- ❖ Η ιδιότητα lang ταυτοποιεί την φυσική γλώσσα των περιεχομένων του στοιχείου, η οποία μπορεί να είναι διαφορετική από το υπόλοιπο έγγραφο. Για παράδειγμα, μέσα σε ένα έγγραφο μπορεί να υπάρχει το κείμενο:

<p>Τι να κάνουμε; c'est la vie, όπως λένε και οι Γάλλοι.</p>

Αναφορές οντοτήτων χαρακτήρων

Από την έκδοση 4.0 και μετά, στην HTML ορίζεται ένα σύνολο από 252 αναφορές οντοτήτων χαρακτήρων και ένα σύνολο από 1.114.050 αναφορές οντοτήτων αριθμών. Και τα δύο σύνολα επιτρέπουν τη γραφή μοναδικών χαρακτήρων ως σήμανση, αντί χρησιμοποιώντας τους ίδιους τους χαρακτήρες. Ένας χαρακτήρας και η αντίστοιχη σήμανση γι' αυτόν θεωρούνται ισοδύναμες οντότητες, και εμφανίζονται ίδια.

Αυτή η δυνατότητα διαφυγής χαρακτήρων με αυτόν τον τρόπο επιτρέπει την ερμηνεία χαρακτήρων όπως < και & (όταν γράφονται ως < και &, αντίστοιχα) ως κείμενο και όχι ως σήμανση. Για παράδειγμα, ο χαρακτήρας < κανονικά σημαίνει την έναρξη μιας ετικέτας, και το & σημαίνει την έναρξη μιας αναφοράς οντότητας χαρακτήρα ή αριθμού. Η γραφή τους ως & ή & ή & επιτρέπει στο & να συμπεριλαμβάνεται μέσα στο περιεχόμενο ενός στοιχείου ή μέσα στην τιμή μιας ιδιότητας. Επίσης, ο χαρακτήρας διπλών εισαγωγικών ("), ο οποίος χρησιμοποιείται για να περιβάλλει την τιμή μιας ιδιότητας, πρέπει να γράφεται ως " ή " ή " όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Ομοίως, ο χαρακτήρας μονών εισαγωγικών ('), πρέπει να γράφεται ως ' ή ' (αλλά όχι ως ', με εξαίρεση την γραφή σε XHTML) όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Αν ένας δημιουργός εγγράφου HTML παραβλέψει να γράψει αυτά τα σύμβολα ως οντότητες χαρακτήρων ή αριθμών, τότε μερικοί browser μπορούν να είναι αρκετά επιεικείς και προσπαθούν να χρησιμοποιήσουν το παραπλήσιο περιεχόμενο για να μαντέψουν την πρόθεση του χρήστη. Το αποτέλεσμα δεν παύει να είναι άκυρη σήμανση, η οποία κάνει το έγγραφο λιγότερο προσιτό σε άλλους browser οι οποίοι μπορεί να προσπαθήσουν να αναλύσουν το έγγραφο για άλλους σκοπούς, όπως για παράδειγμα για την καταχώρηση του περιεχομένου του σε μηχανές αναζήτησης.

Η διαφυγή διευκολύνει επίσης τη χρήση χαρακτήρων που είτε είναι δύσκολο να δακτυλογραφηθούν, είτε δεν συμπεριλαμβάνονται καν στην κωδικοποίηση χαρακτήρων του εγγράφου. Για παράδειγμα, ο τονισμένος χαρακτήρας e (δηλαδή το é), ο οποίος τυπικά συναντάται σε μερικές ευρωπαϊκές γλώσσες, μπορεί να γραφεί σε οποιοδήποτε έγγραφο HTML ως η οντότητα é ή ως η αριθμητική αναφορά é ή é, χρησιμοποιώντας χαρακτήρες οι οποίοι είναι διαθέσιμοι σε όλα τα πληκτρολόγια και υποστηρίζονται σε όλες τις κωδικοποιήσεις χαρακτήρων. Οι κωδικοποιήσεις Unicode, όπως η UTF-8, είναι συμβατές με όλους τους σύγχρονους

browser και επιτρέπουν την πρόσβαση σε σχεδόν όλους τους χαρακτήρες όλων των συστημάτων γραφής του κόσμου.

Τύποι δεδομένων

Στην HTML ορίζονται μερικοί τύποι δεδομένων για το περιεχόμενο των στοιχείων, όπως σενάρια εντολών ή stylesheet, και μια πληθώρα τύπων για τις τιμές των ιδιοτήτων, συμπεριλαμβανομένων των ID, των name, των URI, διαφόρων αριθμών και μονάδων μήκους, γλωσσών, τύπων αρχείων πολυμέσων, χρωμάτων, κωδικοποιήσεων χαρακτήρων, ημερομηνιών κλπ.

Δήλωση τύπου εγγράφου

Τα έγγραφα HTML πρέπει να αρχίζουν με μια Δήλωση τύπου εγγράφου (Document Type Declaration, ανεπίσημα λέγεται και «doctype»). Αυτή η δήλωση βοηθά τους browser να καταλάβουν πώς πρέπει να διαβάσουν το περιεχόμενο του εγγράφου και πώς να το παρουσιάσουν μετά, και ιδιαίτερα όταν χρησιμοποιείται το quirks mode.

Ο αρχικός σκοπός του doctype ήταν να επιτρέψει την ανάλυση και επιβεβαίωση των εγγράφων HTML από εργαλεία SGML τα οποία ήταν βασισμένα στο Document Type Definition (DTD). Το DTD στο οποίο αναφέρεται το DOCTYPE περιέχει μια γραμματική σε γλώσσα μηχανής, η οποία καθορίζει τι επιτρέπεται και τι απαγορεύεται να υπάρχει μέσα στο έγγραφο. Οι browser, από την άλλη, δεν υλοποιούν την HTML ως εφαρμογή της SGML και συνεπώς δεν διαβάζουν το DTD. Στην HTML5 δεν ορίζεται κανένα DTD, λόγω εγγενών τεχνολογικών περιορισμών. Έτσι το doctype `<!doctype html>` δεν αναφέρεται σε κανένα.

Σημασιολογική

Η Σημασιολογική HTML είναι ένας τρόπος γραφής της HTML με έμφαση στο νόημα της σημασμένης πληροφορίας παρά στην εμφάνισή της. Η HTML συμπεριλαμβάνει σημασιολογικά στοιχεία από την σύλληψή της, συμπεριλαμβάνει όμως και στοιχεία αποκλειστικά εμφανισιακά, όπως τις ετικέτες ``, `<i>` και `<center>`. Υπάρχουν επίσης και τα σημασιολογικά ουδέτερα `span` και `div`. Από τα τέλη της δεκαετίας του 1990, όταν τα CSS άρχισαν να δουλεύουν στους περισσότερους browser, οι σχεδιαστές ιστοσελίδων ενθαρρύνονταν να αποφεύγουν τη χρήση των εμφανισιακών ετικετών της HTML και να προτιμούν να διαχωρίζουν το περιεχόμενο από την παρουσίαση.

Σε μια συζήτηση για τον Σημασιολογικό ιστό, ο Τιμ Μπέρνερς Λι και άλλοι έδωσαν παραδείγματα τρόπων με τους οποίους έξυπνα λογισμικά πελάτη ιστού θα μπορούν μια μέρα να περιηγούνται στον Ιστό και να βρίσκουν, να φιλτράρουν και να συσχετίζουν τα προηγουμένως άσχετα και αδημοσίευτα δεδομένα, προς όφελος των ανθρώπων χρηστών. Δέκα χρόνια μετά, δεν είναι ακόμα κοινή η χρήση τέτοιων προγραμμάτων, αλλά μερικές από τις ιδέες του Web 2.0, των mashups και των ιστοσελίδων σύγκρισης

τιμών προϊόντων μπορεί να πλησιάζουν στο σκεπτικό εκείνο. Η βασική διαφορά μεταξύ αυτών των εφαρμογών και των έξυπνων λογισμικών του Μπέρνερς Λι, είναι ότι οι τρέχουσες μέθοδοι συγκέντρωσης πληροφοριών συνήθως σχεδιάζεται από προγραμματιστές ιστού, οι οποίοι ήδη ξέρουν τις τοποθεσίες ιστού και τα API των δεδομένων που θέλουν να συνενώσουν, να συγκρίνουν και να συνδυάσουν.

Ένας σημαντικός τύπος λογισμικοί που περιηγείται αυτόματα το διαδίκτυο και διαβάζει ιστοσελίδες, χωρίς προηγούμενη γνώση του περιεχομένου τους, είναι και τα προγράμματα των μηχανών αναζήτησης. Αυτά τα λογισμικά εξαρτώνται από την σημασιολογική σαφήνεια των ιστοσελίδων που βρίσκουν, καθώς χρησιμοποιούν διάφορες τεχνικές και αλγόριθμους για να διαβάζουν και να ταξινομούν εκατομμύρια σελίδων καθημερινά, και να παρέχουν στους επισκέπτες τους τη δυνατότητα αναζήτησης, χωρίς την οποία ο Παγκόσμιος Ιστός θα είχε μόνο ένα μικρό κλάσμα της χρησιμότητάς του.

Οι σημασιολογικές δομές που ήδη υπάρχουν στην HTML, είναι σημαντικό να εφαρμόζονται καθολικά, για να βοηθούν το περιεχόμενο του δημοσιευμένου κειμένου να γίνεται καλύτερα αντιληπτό. Με αυτό τον τρόπο, τα λογισμικά των μηχανών αναζήτησης, αλλά και άλλα λογισμικά που φτιάχνουν mashup ή άλλα υβρίδια από το περιεχόμενο των ιστότοπων θα μπορούν καλύτερα να εκτιμούν τη σημασία του κειμένου που βρίσκουν στα έγγραφα HTML.[56] Προς το σκοπό αυτό, οι ετικέτες της HTML που εξυπηρετούσαν μόνο εμφανισιακό σκοπό θεωρούνται ξεπερασμένες στην XHTML, και απαγορεύονται στην HTML5.

Τέλος, η σημασιολογικά γραμμένη HTML βελτιώνει την προσιτότητα των εγγράφων στον ιστό. Για παράδειγμα, στους browser για χρήστες με προβλήματα όρασης ή ακοής, είναι πιο εύκολη η σωστή απόδοση της δομής του εγγράφου με τρόπο άλλο από τον οπτικό, όταν αυτό είναι σωστά σημασιολογικά σημασμένο.

Μεταφορά της HTML

Τα αρχεία HTML, όπως συμβαίνει και με τους υπόλοιπους τύπους αρχείων του υπολογιστή, μπορούν να μεταφερθούν με πολλούς τρόπους. Ωστόσο, λόγω της λειτουργίας της HTML, οι δύο πιο συνηθισμένοι τρόποι είναι μέσω HTTP από έναν εξυπηρετητή, ή μέσω E-mail.

HTTP

Ο Παγκόσμιος Ιστός αποτελείται κυρίως από αρχεία HTML τα οποία μεταφέρονται από εξυπηρετητές προς browsers χρησιμοποιώντας το πρωτόκολλο HTTP. Ωστόσο, μια που το ίδιο πρωτόκολλο μπορεί επιπλέον να μεταφέρει εικόνες, ήχο και άλλο περιεχόμενο, χρειάζεται ένας τρόπος αναγνώρισης του τύπου του περιεχομένου που μεταφέρεται. Έτσι, μαζί με το ίδιο το αρχείο, μεταφέρονται και μερικές επιπλέον

πληροφορίες, ή μεταδεδομένα, μεταξύ αυτών και ο τύπος MIME (για παράδειγμα text/html ή application/xhtml+xml) καθώς και η κωδικοποίηση χαρακτήρων που χρησιμοποιείται.

Στους σύγχρονους browser, ο τύπος MIME που συνοδεύει το αρχείο HTML μπορεί να επηρεάζει τον τρόπο που αυτό εμφανίζεται. Για παράδειγμα, ένα αρχείο το οποίο συνοδεύεται από έναν τύπο XHTML MIME, αναμένεται να είναι γραμμένο σε γλώσσα σωστά διατυπωμένη, κατά τα πρότυπα της XML. Αν δεν είναι, τότε τα σφάλματα στον κώδικα μπορεί να αποτρέψουν την ορθή απεικόνιση του αρχείου. Επειδή η XHTML 1.0 και η XHTML 1.1 είναι πάντα συμβατές με την XML, το W3C υποστηρίζει ότι δεν παίζει ρόλο ποιος τύπος MIME θα χρησιμοποιηθεί.

Ονόματα αρχείων

Ο πιο κοινός τύπος αρχείου για έγγραφα HTML είναι .html, όμως έχει επιβιώσει και η συντόμευση .htm, από μερικά παλαιότερα λειτουργικά συστήματα που δεν αναγνώριζαν επεκτάσεις αρχείων με περισσότερα από τρία γράμματα. Επιπλέον, στα Microsoft Windows χρησιμοποιείται ο τύπος .hta (από το HTML Application) ο οποίος δείχνει ότι το αρχείο συμπεριλαμβάνει HTML μαζί με κάποια δυναμικά στοιχεία που το κάνουν να εκτελείται ως εφαρμογή.

Δημιουργία με προγράμματα WYSIWYG

Ένα θέμα αντιπαράθεσης στις κοινότητες των δημιουργών περιεχομένου για τον Ιστό είναι η χρήση προγραμμάτων WYSIWYG (What You See Is What You Get), δηλαδή προγραμμάτων στα οποία ο χρήστης σχεδιάζει οπτικά τη σελίδα όπως θα φαίνεται μετά στον browser του επισκέπτη. Αυτό κάνει τη γνώση της HTML προαιρετική για την κατασκευή μιας ιστοσελίδας. Αυτό το μοντέλο δημιουργίας με προγράμματα WYSIWYG έχει γνωρίσει έντονες κριτικές, [60][61] κυρίως λόγω της κακής ποιότητας του κώδικα που δημιουργείται αυτόματα.

Φόρμες

Αν και ο κύριος σκοπός μιας ιστοσελίδας είναι η δημοσίευση πληροφοριών στο Internet, δεν είναι ο μοναδικός. Όλο και περισσότερα sites χρησιμοποιούν την αμφίδρομη επικοινωνία, δηλαδή δεν παρέχουν απλά πληροφορίες προς τους επισκέπτες, αλλά ζητούν από αυτούς να επιλέγουν ή να πληκτρολογούν στοιχεία. Για παράδειγμα, πολλά sites περιέχουν polls τα οποία παρουσιάζουν μια ερώτηση και ζητούν από τον επισκέπτη να επιλέξει μια από τις απαντήσεις. Στην περίπτωση αυτή έχουμε αμφίδρομη επικοινωνία, καθώς η ροή της πληροφορίας δεν γίνεται μόνο από τον Server που φιλοξενεί την ιστοσελίδα προς τον επισκέπτη αλλά και το αντίθετο.

Οι Φόρμες μας δίνουν την δυνατότητα να μετασχηματίσουμε την ιστοσελίδα μας από μια μάζα πληροφοριών που ο επισκέπτης διαβάζει παθητικά σε μια αμφίδρομη

επικοινωνία όπου έχει έναν πιο ενεργητικό ρόλο πληκτρολογώντας στοιχεία ή κάνοντας κλικ με το ποντίκι του σε διάφορες επιλογές. Εδώ παρουσιάζεται μια φόρμα στην οποία ο επισκέπτης πληκτρολογεί στοιχεία. Τέτοιες φόρμες συνήθως χρησιμοποιούνται για την εγγραφή μελών σε κάποια ψηφιακή κοινότητα (όπως forums, email lists κτλ.)

CSS (Cascading Style Sheets)

Η CSS (Cascading Style Sheets – διαδοχικά φύλλα ύφους ή επάλληλα φύλλα ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης.

Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Η αλληλουχία εφαρμογής των φύλλων ύφους

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

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

Συνεπώς για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες. Το πρότυπο css για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία – σειρά – στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί πχ η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της αλληλουχίας – σειράς – είναι ο ακόλουθος:

- ❖ Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέξει (ταιριάζει με αυτό).
- ❖ Ταξινόμηση με βάση τη σημασία (κανονική ή σημαντική) και προέλευση (συγγραφέας, χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:
 - Δηλώσεις πλοηγού χρήστη
 - Κανονικές δηλώσεις χρήστη
 - Κανονικές δηλώσεις συγγραφέα
 - Σημαντικές δηλώσεις συγγραφέα
 - Σημαντικές δηλώσεις χρήστη
- ❖ Ταξινόμηση τις δηλώσεις ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν των πιο

γενικών. Τα ψευδό-στοιχεία και οι ψευδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.

- ❖ Τέλος ταξινόμησε ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος, προέλευση και εξειδίκευση, αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα ύφους θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο ύφους.

Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων) η δήλωση που θα είναι τελευταία στην σειρά που αναλύθηκε πιο πάνω.

ΚΕΦΑΛΑΙΟ 4- ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΤΟ POWERBUILDER

Το PowerBuilder είναι ένα αντικειμενοστραφές περιβάλλον ανάπτυξης γραφικών εφαρμογών. Χρησιμοποιώντας το PowerBuilder, μπορούμε εύκολα να αναπτύξουμε πολλούς τύπους εφαρμογών και εξαρτημάτων. Το PowerBuilder παρέχει όλα τα εργαλεία που χρειάζονται για την ανάπτυξη συστημάτων επιχειρήσεων, όπως συστήματα εγγραφής, λογιστικής και παραγωγής.

Μια εφαρμογή πελάτη PowerBuilder μπορεί να περιέχει:

- ❖ Ένα μενού αλληλεπίδρασης με τον χρήστη, παράθυρα και στοιχεία ελέγχου παραθύρων με τα οποία οι χρήστες αλληλοεπιδρούν για να κατευθύνουν μια εφαρμογή.
- ❖ Λογική επεξεργασίας εφαρμογών Σενάρια συμβάντων και λειτουργιών στις οποίες κωδικοποιείτε επιχειρηματικούς κανόνες, κανόνες επαλήθευσης και άλλη επεξεργασία εφαρμογών. Το PowerBuilder σας επιτρέπει να κωδικοποιείτε λογική επεξεργασίας εφαρμογών ως μέρος του περιβάλλοντος χρήστη ή σε ξεχωριστές ενότητες που ονομάζονται αντικείμενα χρήστη προσαρμοσμένης κλάσης.

Σε μια πολύπλοκη εφαρμογή, οι ενότητες που περιέχουν λογική επεξεργασίας εφαρμογών (που αναπτύσσετε σε ένα διακομιστή) ονομάζονται συστατικά. Μπορούμε να σχεδιάσουμε, να δημιουργήσουμε και να αναπτύξουμε αντικείμενα χρήστη προσαρμοσμένης κλάσης ως συστατικά στοιχεία διακομιστή εφαρμογών.

Οι εφαρμογές PowerBuilder και ο χρήστης

Σε μια εφαρμογή, οι χρήστες ελέγχουν τι συμβαίνει με τις ενέργειες που λαμβάνουν. Για παράδειγμα, όταν ένας χρήστης κάνει κλικ σε ένα κουμπί, επιλέγει ένα στοιχείο από ένα μενού ή εισάγει δεδομένα σε ένα πλαίσιο κειμένου, ένα ή περισσότερα συμβάντα ενεργοποιούνται. Μπορούμε να γράψουμε σενάρια που καθορίζουν την επεξεργασία που θα πρέπει να συμβεί όταν ενεργοποιούνται τα συμβάντα.

Τα Windows, τα στοιχεία ελέγχου και τα άλλα στοιχεία εφαρμογής που δημιουργείτε με το PowerBuilder διαθέτουν ένα σύνολο προκαθορισμένων συμβάντων. Για παράδειγμα, κάθε κουμπί έχει ένα συμβάν Clicked που σχετίζεται με αυτό και κάθε πλαίσιο κειμένου έχει τροποποιημένο συμβάν. Τις περισσότερες φορές, τα προκαθορισμένα γεγονότα είναι όλα που χρειάζεστε. Ωστόσο, σε ορισμένες περιπτώσεις, μπορεί να θέλετε να ορίσετε τα δικά σας γεγονότα.

Η γλώσσα PowerScript

Μπορείτε να γράψετε σενάρια χρησιμοποιώντας τη γλώσσα PowerBuilder που ονομάζεται PowerScript. Οι κώδικες αποτελούνται από εντολές PowerScript, λειτουργίες και δηλώσεις που εκτελούν επεξεργασία ως απάντηση σε ένα συμβάν. Για παράδειγμα, η δέσμη ενεργειών για συμβάν με ένα κλικ ενός κουμπιού ενδέχεται να ανακτήσει και να εμφανίσει πληροφορίες από τη βάση δεδομένων. Το σενάριο για ένα τροποποιημένο συμβάν πλαισίου κειμένου ενδέχεται να αξιολογήσει τα δεδομένα και να εκτελέσει επεξεργασία με βάση τα δεδομένα. Η εκτέλεση μιας δέσμης ενεργειών μπορεί επίσης να προκαλέσει την ενεργοποίηση άλλων συμβάντων. Για παράδειγμα, το σενάριο για ένα συμβάν Clicked σε ένα κουμπί μπορεί να ανοίξει ένα άλλο παράθυρο, ενεργοποιώντας το ανοιχτό συμβάν σε αυτό το παράθυρο.

Λειτουργίες PowerScript

Το PowerScript παρέχει μια πλούσια ποικιλία ενσωματωμένων λειτουργιών που μπορούν να λειτουργήσουν στα διάφορα στοιχεία της εφαρμογής σας. Για παράδειγμα, υπάρχει μια λειτουργία για να ανοίξετε ένα παράθυρο, μια λειτουργία για κλείσιμο ενός παραθύρου, μια λειτουργία για ενεργοποίηση ενός κουμπιού, μια λειτουργία ενημέρωσης της βάσης δεδομένων κ.ο.κ. Έχει επίσης την δυνατότητα να δημιουργήσουμε τις δικές μας λειτουργίες για να ορίσουμε την επεξεργασία που είναι μοναδική στην εφαρμογή μας.

Αντικειμενοστραφής προγραμματισμός με το PowerBuilder

Κάθε μενού ή παράθυρο που δημιουργείτε με το PowerBuilder είναι μια αυτοτελής ενότητα που ονομάζεται αντικείμενο. Τα βασικά δομικά στοιχεία μιας εφαρμογής PowerBuilder είναι τα αντικείμενα που δημιουργείτε. Κάθε αντικείμενο περιέχει τα ιδιαίτερα χαρακτηριστικά και συμπεριφορές (ιδιότητες, συμβάντα και λειτουργίες) που είναι κατάλληλα γι' αυτό. Αξιοποιώντας τεχνικές προγραμματισμού αντικειμενοστρεφούς προγραμματισμού, όπως ενθυλάκωση, κληρονομιά και πολυμορφισμό, μπορείτε να αξιοποιήσετε στο έπακρο κάθε αντικείμενο που δημιουργείτε, κάνοντας την εργασία σας πιο επαναχρησιμοποιήσιμη, επεκτάσιμη και ισχυρή.

.NET εφαρμογές

Εάν χρησιμοποιείτε την έκδοση Enterprise του PowerBuilder, μπορείτε να αναπτύξετε εφαρμογές και στοιχεία για το περιβάλλον .NET. Αυτές περιλαμβάνουν εφαρμογές Web Forms που αναπτύσσετε σε διακομιστές IIS χρησιμοποιώντας εφαρμογές ASP.NET και Windows Forms που μπορείτε να αναπτύξετε με λειτουργίες smart client. Μπορείτε επίσης να μετατρέψετε αντικειμενικά αντικείμενα μη προσαρμοσμένων κλάσεων απευθείας σε συναρμολογήσεις .NET και μπορείτε να τα αναπτύξετε ως υπηρεσίες .NET Web.

Πολλαπλές εφαρμογές

Το PowerBuilder σας επιτρέπει να δημιουργείτε εφαρμογές που εκτελούνται σε περιβάλλον κατανεμημένων υπολογιστών. Μια εφαρμογή multitier σας επιτρέπει:

- ❖ Κεντράρετε την επιχειρησιακή λογική σε διακομιστές, όπως EAServer, JBoss, WebLogic, WebSphere ή COM +
- ❖ Λειτουργίες εφαρμογής διαμερισμάτων μεταξύ του πελάτη και του διακομιστή, μειώνοντας έτσι το φόρτο εργασίας του πελάτη
- ❖ Κατασκευάστε επεκτάσιμες εφαρμογές που είναι εύκολο να συντηρηθούν

Για πληροφορίες σχετικά με εφαρμογές multitier, ανατρέξτε στις ενότητες σχετικά με τις τεχνικές κατανεμημένης εφαρμογής στις τεχνικές εφαρμογής.

Σύνδεση βάσης δεδομένων

Το PowerBuilder παρέχει εύκολη πρόσβαση σε εταιρικές πληροφορίες που είναι αποθηκευμένες σε μια μεγάλη ποικιλία βάσεων δεδομένων. Τα δεδομένα μπορούν να αποκτήσουν πρόσβαση μέσω των διασυνδέσεων PowerBuilder ODBC ή JDBC, μέσω διακομιστή πρόσβασης δεδομένων μεσαίου επιπέδου όπως ο διακομιστής Sybase DirectCONNECT™ ή μέσω εγγενής ή άμεσης σύνδεσης σε μια βάση δεδομένων.

Σε πολλούς οργανισμούς, οι ειδικοί της βάσης δεδομένων διατηρούν τη βάση δεδομένων. Ωστόσο, για να εκμεταλλευτείτε πλήρως το PowerBuilder, πρέπει να ξέρετε πώς να εργαστείτε με βάσεις δεδομένων.

Ορισμός πηγής δεδομένων(data source)

Χρησιμοποιώντας το διαχειριστή ODBC ή άλλα βοηθητικά προγράμματα σύνδεσης βάσεων δεδομένων, μπορείτε να ορίσετε μια βάση δεδομένων ως πηγή δεδομένων για την εφαρμογή σας. Μπορείτε να αποκτήσετε πρόσβαση στο διαχειριστή ODBC από το παράθυρο διαλόγου Προφίλ δεδομένων. Οι ορισμοί των πηγών δεδομένων ODBC αποθηκεύονται στο κλειδί μητρώου odbc.ini.

Χρήση προφίλ βάσεων δεδομένων για σύνδεση

Αφού καθορίσετε μια πηγή δεδομένων, μπορείτε να δημιουργήσετε ένα προφίλ βάσης δεδομένων για αυτήν. Ένα προφίλ βάσης δεδομένων είναι ένα σύνολο παραμέτρων που ορίζει μια σύνδεση με μια συγκεκριμένη πηγή δεδομένων ή βάση δεδομένων. Τα προφίλ βάσεων δεδομένων παρέχουν έναν εύκολο τρόπο για να διαχειρίζεστε τις συνδέσεις βάσεων δεδομένων που χρησιμοποιείτε συχνά. Όταν αναπτύσσετε μια εφαρμογή, μπορείτε να αλλάξετε προφίλ βάσεων δεδομένων για να συνδεθείτε σε διαφορετική πηγή δεδομένων.

Συνδέσεις βάσης δεδομένων

Το PowerBuilder μπορεί να δημιουργήσει μια σύνδεση με τη βάση δεδομένων είτε στο περιβάλλον σχεδιασμού είτε στο χρόνο εκτέλεσης. Το PowerBuilder συνδέεται σε μια βάση δεδομένων όταν ανοίγετε ορισμένους ζωγράφους όταν συντάσσετε ή αποθηκεύετε μια δέσμη ενεργειών PowerBuilder που περιέχει ενσωματωμένες εντολές SQL ή όταν εκτελείτε μια εφαρμογή PowerBuilder που έχει πρόσβαση στη βάση δεδομένων.

Για να διατηρήσουμε τους ορισμούς των βάσεων δεδομένων με το PowerBuilder, χρησιμοποιούμε τον παράθυρο σχεδίασης βάσης δεδομένων. Το παράθυρο αυτό μας επιτρέπει να:

- ❖ Δημιουργήσουμε, τροποποιήσουμε και διαγράψουμε πίνακες
- ❖ Δημιουργήσουμε, τροποποιήσουμε και διαγράψουμε τα κύρια και ξένα πλήκτρα
- ❖ Δημιουργήσουμε και διαγράψουμε ευρετήρια(indexes)
- ❖ Ορίσουμε και τροποποιήσουμε τα εκτεταμένα χαρακτηριστικά για τις στήλες
- ❖ Διαγραφή προβολών (views)

Το περιβάλλον του PowerBuilder

Workspaces and εφαρμογές

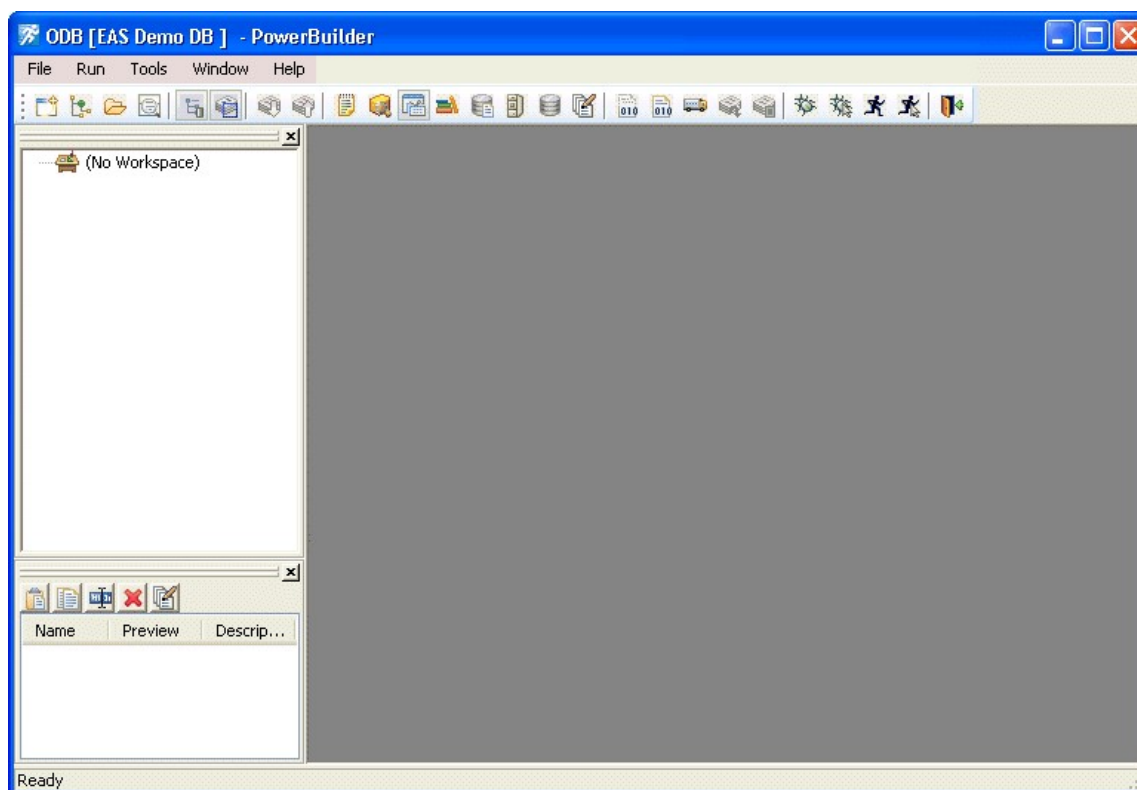
Στο PowerBuilder, εργάζεστε με μια ή περισσότερες εφαρμογές σε ένα χώρο εργασίας. Μπορείτε να προσθέσετε όσες εφαρμογές στο χώρο εργασίας θέλετε, να ανοίξετε και να επεξεργαστείτε αντικείμενα σε πολλούς στόχους και να δημιουργήσετε και να αναπτύξετε πολλαπλούς στόχους ταυτόχρονα.

Μια εφαρμογή PowerBuilder μπορεί να ένας από τους διάφορους τύπους:

- ❖ Μια εκτελέσιμη εφαρμογή πελάτη / διακομιστή.
- ❖ Μια .NET εφαρμογή που μπορούμε να χρησιμοποιήσουμε για να αναπτύξουμε εφαρμογές ως .NET Windows ή ASP.NET εφαρμογές ή να αναπτύξουμε μη εμφανής προσαρμοσμένες κλάσεις συστατικά ως συναρμολογήσεις .NET ή υπηρεσίες Web.
- ❖ Μια εφαρμογή EAServer ή μια εφαρμογή διακομιστή. Η εφαρμογής μπορεί να αναπτυχθεί στο EAServer ή σε άλλο διακομιστή συμβατό με J2EE.

Το περιβάλλον ανάπτυξης

Όταν ανοίγει το PowerBuilder, ανοίγει ένα παράθυρο που περιέχει μια γραμμή μενού και το PowerBar στην κορυφή και τα παράθυρα Systems Tree και Clip στα αριστερά.



Δέντρο συστήματος (System Tree)

Το παράθυρο System Tree μπορεί να χρησιμεύσει ως ο κόμβος των αναπτυξιακών δραστηριοτήτων. Μπορούμε να το χρησιμοποιήσουμε για να ανοίξουμε, να εκτελέσουμε, να εντοπίσουμε σφάλματα και να δημιουργήσουμε τους στόχους σας, καθώς και για προγραμματισμό drag-and-drop.

Παράθυρο Clip

Το παράθυρο Clip σας επιτρέπει να αποθηκεύετε τα αποσπάσματα κώδικα που χρησιμοποιείτε συχνά.

Παράθυρο εξόδου

Η έξοδος μιας ποικιλίας λειτουργιών (μετάβαση, δημιουργία, ανάπτυξη, εκτέλεση έργου, αποθήκευση αντικειμένου και αναζητήσεις) εμφανίζεται σε παράθυρο εξόδου στο κάτω μέρος του κύριου παραθύρου. Το παράθυρο εξόδου ανοίγει αυτόματα όταν παράγονται πληροφορίες εξόδου, αλλά μπορείτε να ανοίξετε το παράθυρο εξόδου οποιαδήποτε στιγμή κάνοντας κλικ στο κουμπί της γραμμής εργαλείων παραθύρου εξόδου.

Εργαλεία σχεδίασης

Μόλις δημιουργήσετε έναν χώρο εργασίας και έναν στόχο PowerScript, δημιουργείτε τα στοιχεία του στόχου χρησιμοποιώντας εργαλεία σχεδίασης. Τα εργαλεία σχεδίασης παρέχονται για την ενίσχυση και την τελειοποίηση των αντικειμένων σε μια εφαρμογή.

Το PowerBuilder παρέχει έναν πακέτο εργαλείων σχεδίασης για κάθε τύπο αντικειμένου που δημιουργούμε. Για παράδειγμα, δημιουργούμε ένα παράθυρο στο παράθυρο σχεδίασης. Εκεί καθορίζετε τις ιδιότητες του παραθύρου και προσθέτετε στοιχεία ελέγχου, όπως κουμπιά και πλαίσια κειμένου.

Έλεγχοι σχεδιασμού-χρόνου

Τα στοιχεία ελέγχου χρόνου σχεδιασμού (DTC) δημιουργούν βασικά HTML και δέσμες ενεργειών από τις πληροφορίες που παρέχετε στα φύλλα ιδιοτήτων. Τα φύλλα ιδιοτήτων εμφανίζονται όταν αποθέτετε έναν κωδικό DTC σε μια ιστοσελίδα στο πρόγραμμα επεξεργασίας HTML.

Το DTC Web DataWindow DTC παρέχει έναν εύκολο τρόπο πρόσβασης σε μια βάση δεδομένων από μια ιστοσελίδα. Εμφανίζει περιεχόμενο δυναμικής βάσης δεδομένων σε μια ποικιλία μορφών παρουσίασης και υποστηρίζει ένθετα, ενημερώσεις και διαγραφές κατά της βάσης δεδομένων.

Λίστα υποχρεώσεων(To do list)

Η Λίστα υποχρεώσεων εμφανίζει μια λίστα με τις εργασίες ανάπτυξης που πρέπει να κάνετε για τον τρέχοντα στόχο. Μπορούμε να δημιουργήσουμε αυτόματα τις καταχωρήσεις στη λίστα "Ολοκλήρωση" από τα περισσότερα παράθυρα του PowerBuilder. Μπορούμε επίσης να πληκτρολογήσουμε καταχωρήσεις ή να τις εισαγάγετε από ένα αρχείο κειμένου και στη συνέχεια να τις συνδέσουμε σε μια εργασία που θέλουμε να ολοκληρώσουμε.

Browser

Ο Browser σάς επιτρέπει να δείτε όλα τα αντικείμενα, τις μεθόδους, τις μεταβλητές και τις δομές που έχουν καθοριστεί ή είναι διαθέσιμες στο στόχο σας στο PowerScript. Τα αντικείμενα στο πρόγραμμα περιήγησης μπορούν να εμφανιστούν με αλφαβητική ή ιεραρχική σειρά. Το παράθυρο αυτό εμφανίζει μεθόδους με τα πλήρη πρωτότυπα (υπογραφές), τα οποία περιλαμβάνουν τους τύπους δεδομένων όλων των παραδειγμάτων και τις τιμές επιστροφής.

PowerBar

Το PowerBar εμφανίζεται όταν ξεκινάτε μια περίοδο λειτουργίας PowerBuilder. Το PowerBar είναι το κύριο σημείο ελέγχου για την κατασκευή εφαρμογών PowerBuilder. Μπορείτε να χρησιμοποιήσουμε τα κουμπιά New, Inherit ή Open στο PowerBar για να ανοίξετε όλους τα παράθυρα του PowerBuilder. Από το PowerBar, μπορούμε επίσης να ανοίξουμε το πρόγραμμα περιήγησης, να εντοπίσουμε σφάλματα ή να εκτελέσουμε την τρέχουσα εφαρμογή, να δημιουργήσουμε και να αναπτύξετε το χώρο εργασίας.

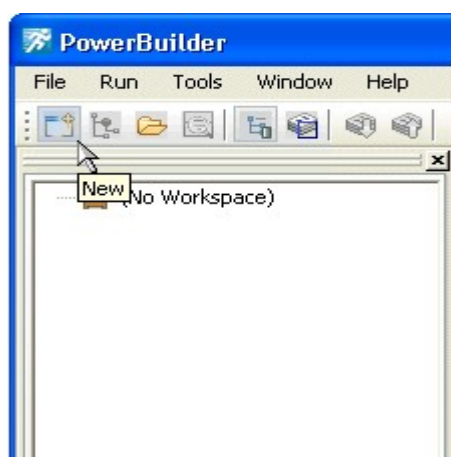
PainterBar

Όταν ανοίγετε έναν παράθυρο σχεδίασης ή ένα πρόγραμμα επεξεργασίας, το PowerBuilder εμφανίζει ένα νέο παράθυρο που έχει ένα χώρο εργασίας στον οποίο σχεδιάζετε το αντικείμενο που δημιουργούμε. Το PowerBuilder εμφανίζει επίσης ένα ή περισσότερα PainterBars με κουμπιά που παρέχουν εύκολη πρόσβαση στα εργαλεία που διαθέτει ο ζωγράφος ή ο συντάκτης. Για παράδειγμα, εδώ είναι το PainterBar για τον ζωγράφο της DataWindow.



PowerTips

Όταν αφήνουμε το δείκτη του ποντικιού πάνω από ένα κουμπί για ένα δευτερόλεπτο ή δύο, το PowerBuilder μπορεί να εμφανίσει μια σύντομη περιγραφή του κουμπιού (PowerTip). Η δυνατότητα εμφάνισης των PowerTips ενεργοποιείται και απενεργοποιείται επιλέγοντας το στοιχείο μενού Εμφάνιση στοιχείων ενέργειας από οποιοδήποτε αναδυόμενο μενού της γραμμής εργαλείων.



Προσαρμογή του περιβάλλοντος

Εκτός από την εμφάνιση κειμένου στα κουμπιά της γραμμής εργαλείων, μπορείτε να μετακινήσετε τις γραμμές εργαλείων γύρω, να προσθέσετε νέες γραμμές εργαλείων και να προσαρμόσετε τις υπάρχουσες. Μπορείτε να προσθέσετε κουμπιά για το άνοιγμα ζωγράφων και την εκτέλεση άλλων δραστηριοτήτων.

Επίσης, μπορείτε να αναδιατάξουμε τις προβολές του συστήματος Tree, Clip και Output, να ορίσουμε προσαρμοσμένες ρυθμίσεις για κάθε παράθυρο, να επιλέξουμε αν

το PowerBuilder θα ανοίξει το τελευταίο σας χώρο εργασίας κατά την εκκίνηση με ή χωρίς παράθυρα , να προσαρμόσουμε τα πλήκτρα συντόμευσης και να αλλάξουμε τα χρώματα και γραμματοσειρές που χρησιμοποιούνται σε δέσμες ενεργειών.

Αντικείμενα του PowerBuilder

Τα βασικά δομικά στοιχεία ενός PowerScript είναι τα εξής αντικείμενα:

Εφαρμογές	Σημείο εισόδου σε μια εφαρμογή
Παράθυρο	Πρωτεύουσα διεπαφή μεταξύ του χρήστη και μιας εφαρμογής PowerBuilder
DataWindow	Ανάκτηση και χειρισμός δεδομένων από σχεσιακή βάση δεδομένων ή άλλη πηγή δεδομένων
Menu	Λίστα εντολών ή επιλογών που μπορεί να επιλέξει ένας χρήστης στο τρέχον ενεργό παράθυρο
Global function	Συνάρτηση που εκτελεί επεξεργασία γενικού σκοπού
Query	Εντολή SQL που χρησιμοποιείται επανειλημμένα ως πηγή δεδομένων για ένα DataWindow
Structure (Δομή)	Συλλογή μιας ή περισσότερων σχετικών μεταβλητών που ομαδοποιούνται με ένα μόνο όνομα
User object	Επαναχρησιμοποιήσιμη μονάδα επεξεργασίας ή σύνολο εντολών, είτε φανερό είτε όχι
Pipeline(Σωλήνωση)	Αναπαράγει δεδομένα σε μια βάση δεδομένων ή από μια βάσεις δεδομένων
Project	Πακέτο εφαρμογών για διανομή στους χρήστες

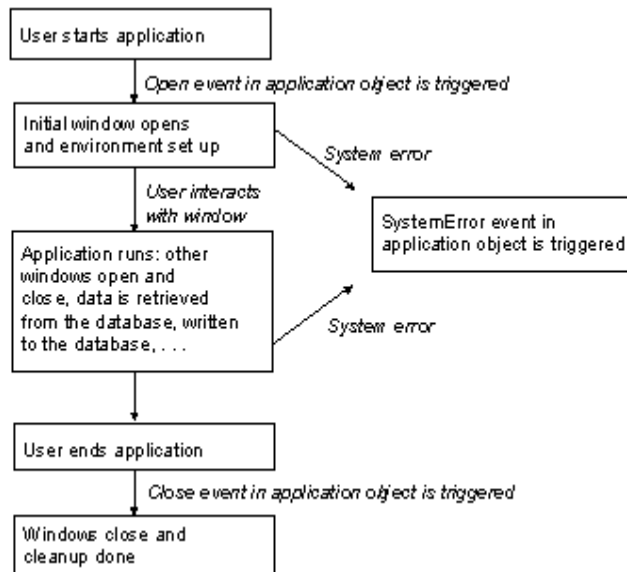
Το αντικείμενο εφαρμογής

Το αντικείμενο εφαρμογής είναι το σημείο εισόδου σε μια εφαρμογή. Είναι ένα διακριτό αντικείμενο που αποθηκεύεται σε μια βιβλιοθήκη PowerBuilder (αρχείο PBL), ακριβώς όπως ένα παράθυρο, ένα μενού, μια λειτουργία ή ένα αντικείμενο DataWindow.

Το αντικείμενο εφαρμογής καθορίζει τη συμπεριφορά σε επίπεδο εφαρμογής, όπως ποιες γραμματοσειρές χρησιμοποιούνται για προεπιλογή για κείμενο και ποια επεξεργασία θα πρέπει να πραγματοποιηθεί όταν αρχίζει και τελειώνει η εφαρμογή.

Όταν ένας χρήστης εκτελεί την εφαρμογή, ενεργοποιείται ένα ανοιχτό συμβάν στο αντικείμενο της εφαρμογής. Το σενάριο που γράφετε για το συμβάν Open ανοίγει τη δραστηριότητα στην εφαρμογή. Όταν ο χρήστης τερματίσει την εφαρμογή, ενεργοποιείται το συμβάν Κλείσιμο στο αντικείμενο εφαρμογής.

Η δέσμη ενεργειών που γράφετε για το συμβάν κλεισίματος συνήθως κάνει όλο τον καθαρισμό που απαιτείται, όπως το κλείσιμο μιας βάσης δεδομένων ή η εγγραφή σε ένα αρχείο προτιμήσεων. Εάν υπάρχουν σοβαρά σφάλματα κατά την εκτέλεση που δεν αλιεύονται με το μηχανισμό χειρισμού εξαιρέσεων του PowerBuilder, ενεργοποιείται το συμβάν SystemError του αντικειμένου εφαρμογής.

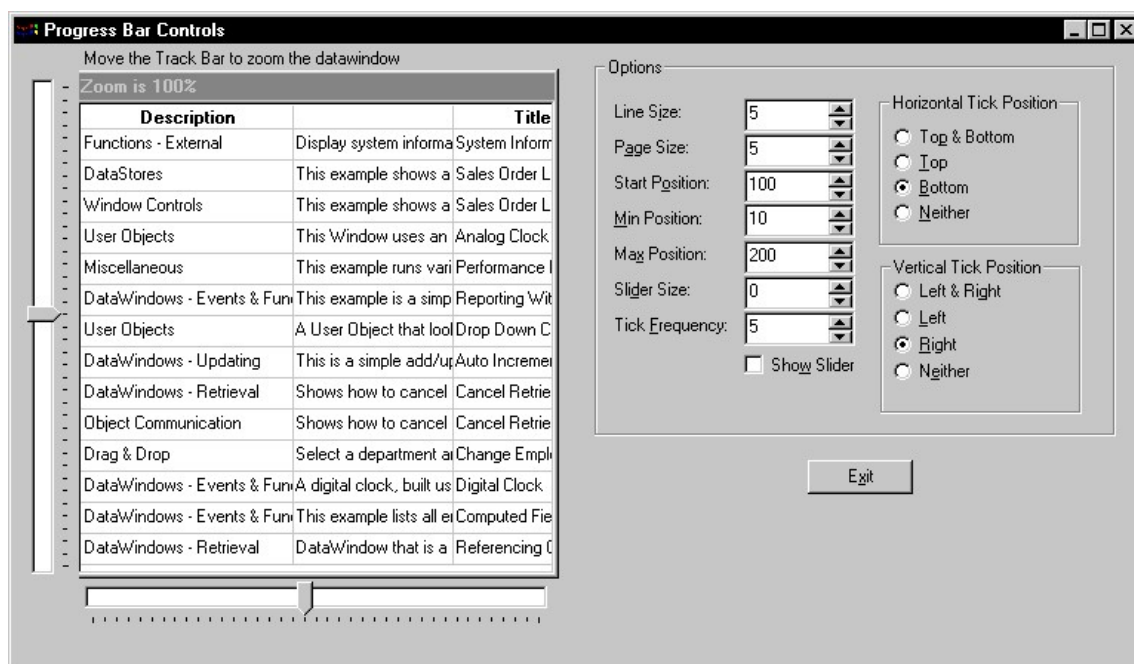


Ο κύκλος ζωής μιας εφαρμογής

Τα παράθυρα είναι η κύρια διεπαφή μεταξύ του χρήστη και μιας εφαρμογής PowerBuilder. Τα παράθυρα μπορούν να εμφανίζουν πληροφορίες, να ζητούν πληροφορίες από ένα χρήστη και να ανταποκρίνονται στις ενέργειες του ποντικιού ή του πληκτρολογίου του χρήστη. Ένα παράθυρο αποτελείται από:

- ❖ Ιδιότητες που καθορίζουν την εμφάνιση και τη συμπεριφορά του παραθύρου (για παράδειγμα, ένα παράθυρο μπορεί να έχει μια γραμμή τίτλου και ένα πλαίσιο Ελαχιστοποίησης)
- ❖ Γεγονότα που ενεργοποιούνται από τις ενέργειες των χρηστών
- ❖ Έλεγχοι που τοποθετούνται στο παράθυρο

Τα παράθυρα μπορούν να έχουν διάφορα είδη στοιχείων ελέγχου, όπως φαίνεται στην παρακάτω εικόνα:



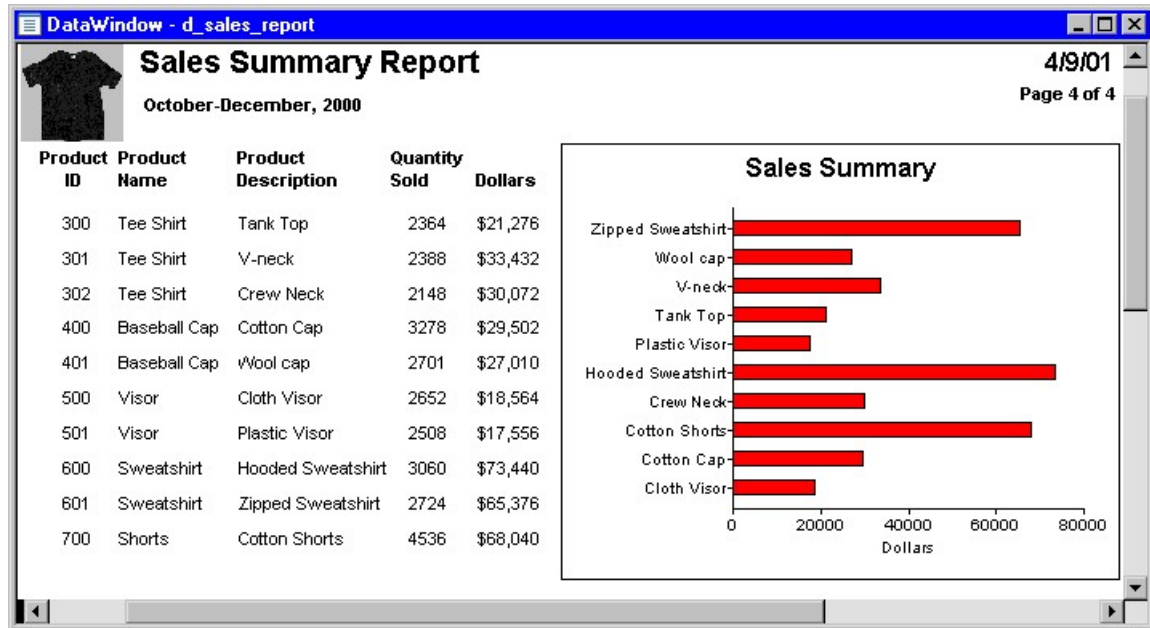
Στα αριστερά του παραθύρου υπάρχει ένα στοιχείο DataWindow με οριζόντιες και κάθετες γραμμές παρακολούθησης. Στα δεξιά υπάρχει ένα πλαίσιο ομάδας που περιέχει στοιχεία ελέγχου στατικού κειμένου (που περιέχουν περιγραφικές ετικέτες), επεξεργάζονται στοιχεία ελέγχου μάσκας (όπως εμφανίζονται όταν είναι ενεργοποιημένη η ιδιότητα SpinControl), πλαίσιο ελέγχου και δύο μικρότερα πλαίσια ομάδας με κουμπιά επιλογής. Στο πλαίσιο της κύριας ομάδας υπάρχει ένα κουμπί εντολής.

Αντικείμενα του DataWindow

Ένα αντικείμενο DataWindow είναι ένα αντικείμενο που χρησιμοποιείτε για να ανακτήσετε και να χειριστείτε δεδομένα από μια σχεσιακή βάση δεδομένων ή άλλη πηγή δεδομένων (όπως ένα φύλλο εργασίας του Excel ή αρχείο dBASE).

Στυλ παρουσίασης Τα DataWindow χειρίζονται επίσης τον τρόπο παρουσίασης των δεδομένων στο χρήστη. Μπορείτε να επιλέξετε από διάφορα στυλ παρουσίασης. Για παράδειγμα, μπορείτε να εμφανίσετε τα δεδομένα σε στυλ πίνακα ή ελεύθερου στυλ.

Υπάρχουν πολλοί τρόποι για να βελτιώσετε την παρουσίαση και τον χειρισμό δεδομένων σε ένα DataWindow. Για παράδειγμα, μπορείτε να συμπεριλάβετε υπολογισμένα πεδία, εικόνες και γραφήματα που συνδέονται άμεσα με τα δεδομένα που ανακτώνται από το DataWindow.



Μορφές εμφάνισης, επεξεργασία στυλ και επικύρωση

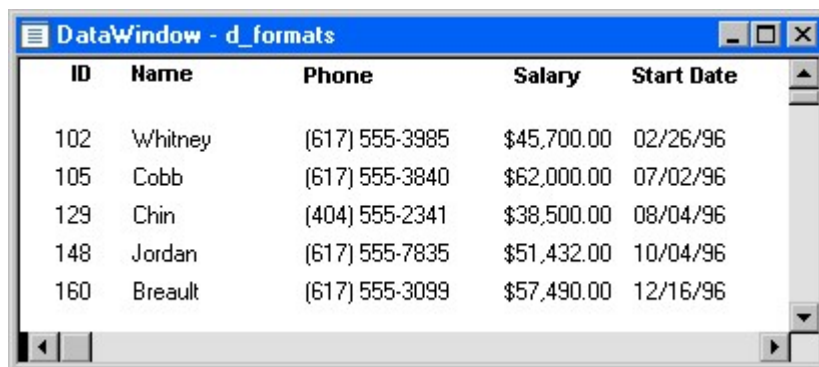
Μπορούμε να καθορίσουμε τον τρόπο εμφάνισης των τιμών για κάθε στήλη και έχουμε την δυνατότητα να επικυρώσουμε δεδομένα που έχουν εισαχθεί από χρήστες σε ένα αντικείμενο DataWindow. Αυτό το κάνουμε καθορίζοντας τις μορφές εμφάνισης, τα στυλ επεξεργασίας και τους κανόνες επικύρωσης για τις στήλες.

Για παράδειγμα, εάν μια στήλη μπορεί να πάρει μόνο ένα μικρό αριθμό αμοιβαία αποκλειστικών τιμών, μπορούμε να έχουμε τα δεδομένα να εμφανίζονται ως κουμπιά επιλογής σε ένα DataWindow ώστε οι χρήστες να γνωρίζουν ποιες είναι οι επιλογές τους.

The screenshot shows a DataWindow window titled "DataWindow - d_edit_styles". The table displays employee information with radio buttons for the "Status" column.

ID	Dept	Name	Status	Salary
102	100	Whitney, Fran	<input checked="" type="radio"/> Active <input type="radio"/> Terminated <input type="radio"/> On Leave	\$45,700.00
105	100	Cobb, Matthew	<input checked="" type="radio"/> Active <input type="radio"/> Terminated <input type="radio"/> On Leave	\$62,000.00
129	200	Chin, Philip	<input checked="" type="radio"/> Active <input type="radio"/> Terminated <input type="radio"/> On Leave	\$38,500.00

Εάν τα δεδομένα περιλαμβάνουν αριθμούς τηλεφώνου, μισθούς και ημερομηνίες, μπορείτε να διαμορφώσετε την οθόνη ώστε να ταιριάζει με τα δεδομένα.



ID	Name	Phone	Salary	Start Date
102	Whitney	(617) 555-3985	\$45,700.00	02/26/96
105	Cobb	(617) 555-3840	\$62,000.00	07/02/96
129	Chin	(404) 555-2341	\$38,500.00	08/04/96
148	Jordan	(617) 555-7835	\$51,432.00	10/04/96
160	Breault	(617) 555-3099	\$57,490.00	12/16/96

Εάν μια στήλη μπορεί να λάβει αριθμούς μόνο σε συγκεκριμένο εύρος, μπορείτε να καθορίσετε έναν απλό κανόνα επικύρωσης για τα δεδομένα. Αυτό μπορεί να σας απαλλάξει από την εγγραφή κώδικα για να βεβαιωθείτε ότι οι χρήστες εισάγουν έγκυρα δεδομένα.

Web DataWindow

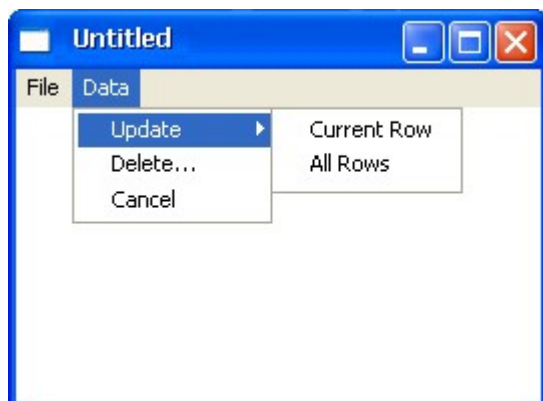
Χρησιμοποιώντας το XML Web DataWindow, το XHTML Web DataWindow ή το Web Web DataWindow, μπορείτε να δημιουργήσετε αντικείμενα DataWindow σε XML (με μεταγενέστερη μετατροπή XSLT σε XHTML), XHTML απευθείας ή HTML και να τα εμφανίσετε σε ένα πρόγραμμα περιήγησης χρησιμοποιώντας ένα εργαλείο PowerBuilder που εκτελείται σε EAServer ή COM + για τη δημιουργία του Web DataWindow.

Menus

Τα μενού είναι λίστες στοιχείων που μπορεί να επιλέξει ένας χρήστης από μια γραμμή μενού για το ενεργό παράθυρο. Τα στοιχεία σε ένα μενού συνήθως σχετίζονται. Παρέχουν στον χρήστη εντολές (όπως το Open και Save As) ή εναλλακτικούς τρόπους εκτέλεσης μιας εργασίας (για παράδειγμα, τα στοιχεία στο μενού Επεξεργασία στα παράθυρα αντιστοιχούν σε κουμπιά στο PainterBar).

Μπορούμε να επιλέξουμε στοιχεία μενού με το ποντίκι ή με το πληκτρολόγιο ή να χρησιμοποιήσουμε τα πλήκτρα επιτάχυνσης (μνημονική πρόσβαση) που ορίζονται για τα στοιχεία. Υπάρχει η δυνατότητα να ορίσουμε τις δικές μας συντομεύσεις πληκτρολογίου για οποιοδήποτε στοιχείο του μενού PowerBuilder από ένα παράθυρο διαλόγου που ανοίγεται με το στοιχείο μενού Εργαλεία> Συντομεύσεις πληκτρολογίου.

Ένα drop down menu είναι ένα μενού κάτω από ένα στοιχείο στη γραμμή μενού. Ένα μενού με κλιμάκωση είναι ένα μενού που εμφανίζεται στο πλάι ενός στοιχείου σε ένα drop down μενού.



Το προηγούμενο παράθυρο εμφανίζει δύο αντικείμενα μενού στη γραμμή (Αρχείο και Δεδομένα), τρία αντικείμενα στο αναπτυσσόμενο μενού Δεδομένα (Ενημέρωση, Διαγραφή και Ακύρωση) και δύο αντικείμενα στο μενού κλιμάκωσης δίπλα στην επιλογή Ενημέρωση “All rows”).

Global Συναρτήσεις

Το PowerBuilder επιτρέπει να ορίσουμε δύο τύπους συναρτήσεων:

- ❖ Οι συναρτήσεις επιπέδου αντικειμένου που έχουν οριστεί για ένα συγκεκριμένο τύπο παραθύρου, μενού ή άλλου τύπου αντικειμένου και έχουν εγκλωβιστεί μέσα στο αντικείμενο για το οποίο έχουν οριστεί. Αυτές διαιρούνται περαιτέρω σε λειτουργίες συστήματος (συναρτήσεις που είναι πάντα διαθέσιμες για αντικείμενα συγκεκριμένης κλάσης αντικειμένων) και λειτουργίες που ορίζονται από το χρήστη.
- ❖ Οι καθολικές συναρτήσεις δεν είναι ενσωματωμένες σε άλλο αντικείμενο, αλλά αποθηκεύονται ως ανεξάρτητα αντικείμενα.

Σε αντίθεση με τις αντικειμενοστραφείς συναρτήσεις, οι γενικές συναρτήσεις δεν δρουν σε συγκεκριμένες περιπτώσεις ενός αντικειμένου. Αντ' αυτού, εκτελούν επεξεργασία γενικού σκοπού, όπως μαθηματικούς υπολογισμούς ή χειρισμούς γραμμών.

Query

Ένα query είναι μια εντολή SQL που αποθηκεύεται με ένα όνομα έτσι ώστε να μπορεί να χρησιμοποιηθεί επανειλημμένα ως πηγή δεδομένων για ένα αντικείμενο DataWindow. Τα ερωτήματα ενισχύουν την παραγωγικότητα των προγραμματιστών, επειδή μπορούν να κωδικοποιηθούν μία φορά αλλά να επαναχρησιμοποιηθούν όσο συχνά χρειάζεται.

Δομές (Structures)

Μια δομή είναι μία συλλογή από μία ή περισσότερες σχετικές μεταβλητές του ίδιου ή διαφορετικού τύπου δεδομένων που ομαδοποιούνται με ένα μόνο όνομα. Οι δομές μας επιτρέπουν να αναφερθούμε σε σχετικές οντότητες ως μονάδες και όχι ως μεμονωμένα. Για παράδειγμα, μπορούμε να ορίσουμε το αναγνωριστικό χρήστη (id), τη διεύθυνση, το επίπεδο πρόσβασης και μια εικόνα (bitmap) του υπαλλήλου ως μια δομή που ονομάζεται `user_struct` και στη συνέχεια να ανατρέξετε σε αυτήν τη συλλογή μεταβλητών ως `user_struct`.

Υπάρχουν δύο είδη δομών:

- ❖ Οι δομές επιπέδου αντικειμένου συσχετίζονται με έναν συγκεκριμένο τύπο αντικειμένου, όπως ένα παράθυρο ή ένα μενού. Αυτές οι δομές μπορούν πάντα να χρησιμοποιηθούν σε σενάρια για το ίδιο το αντικείμενο. Μπορείτε επίσης να επιλέξετε να κάνετε τις δομές προσβάσιμες από άλλα σενάρια.
- ❖ Οι γενικές δομές δεν σχετίζονται με κανένα αντικείμενο ή τύπο αντικειμένου σε μια εφαρμογή. Μπορείτε να δηλώσετε μια εμφάνιση της δομής και να την αναφέρετε σε οποιοδήποτε σενάριο σε μια εφαρμογή.

Αντικείμενα Χρήστη (Users Object)

Οι εφαρμογές συχνά έχουν κοινά χαρακτηριστικά. Για παράδειγμα, πολλές εφαρμογές ενδέχεται να έχουν ένα κουμπί κλεισίματος που εκτελεί ένα ορισμένο σύνολο λειτουργιών και στη συνέχεια κλείνει το παράθυρο ή ενδέχεται να έχουν στοιχεία ελέγχου `DataWindow` που εκτελούν τον ίδιο τύπο ελέγχου σφαλμάτων. Ορισμένες εφαρμογές ενδέχεται να απαιτούν όλα τα τυπικά προγράμματα προβολής αρχείων. Αν διαπίστωσε ότι χρησιμοποιείτε επαναληπμένα την ίδια λειτουργία εφαρμογής, θα πρέπει να ορίσετε ένα αντικείμενο χρήστη. Ορίζετε το αντικείμενο χρήστη μία φορά και το χρησιμοποιείτε όσες φορές χρειάζεστε. Τα αντικείμενα χρήστη μπορούν να είναι φανερά ή μη. Μπορούν να χωριστούν περαιτέρω σε τυπικά ή προσαρμοσμένα αντικείμενα χρήστη. Τα τυπικά αντικείμενα χρήστη, είτε φανερά είτε μη, είναι αντικείμενα συστήματος που είναι πάντα διαθέσιμα με το PowerBuilder. Μπορούμε επίσης να χρησιμοποιήσετε χειριστήρια για εξωτερικά οπτικά αντικείμενα που δημιουργήθηκαν εκτός του PowerBuilder. Οι κύριοι τύποι αντικειμένων χρηστών είναι:

- ❖ **Εμφανή αντικείμενα χρήστη.** Αυτά είναι επαναχρησιμοποιήσιμα στοιχεία ελέγχου ή σύνολα στοιχείων ελέγχου που έχουν συνεπή συμπεριφορά. Για παράδειγμα, ένα οπτικό αντικείμενο χρήστη μπορεί να αποτελείται από πολλά κουμπιά που λειτουργούν ως μονάδα. Τα κουμπιά θα μπορούσαν να έχουν συγγράμματα που συνδέονται με αυτά που εκτελούν τυπική επεξεργασία. Μόλις καθοριστεί το αντικείμενο, μπορείτε να το χρησιμοποιήσετε όσο συχνά θέλετε.
- ❖ **Μη εμφανή αντικείμενα χρήστη.** Αυτές είναι επαναχρησιμοποιήσιμες μονάδες επεξεργασίας που δεν έχουν εμφανή στοιχείο. Τα συνηθισμένα

αντικείμενα χρήστη κατηγορίας κληρονομούν γεγονότα και ιδιότητες από ενσωματωμένα αντικείμενα συστήματος. Συνήθως χρησιμοποιούνται μη εμφανή αντικείμενα για να ορίσουν επιχειρηματικούς κανόνες και άλλες επεξεργασίες που λειτουργούν ως μονάδες.

Για παράδειγμα, μπορεί να θέλετε να υπολογίσετε προμήθειες ή να εκτελέσετε στατιστική ανάλυση σε διάφορες εφαρμογές. Για να γίνει αυτό, μπορείτε να ορίσετε ένα αντικείμενο χρήστη προσαρμοσμένης κλάσης. Για να χρησιμοποιήσετε ένα αντικείμενο προσαρμοσμένης κλάσης, δημιουργείτε μια παρουσία του αντικειμένου σε μια δέσμη ενεργειών και καλείτε τις λειτουργίες του.

Τα αντικείμενα χρήστη προσαρμοσμένης κλάσης, τα οποία ορίζουν λειτουργίες και μεταβλητές, αποτελούν το θεμέλιο των εφαρμογών πολλών εφαρμογών του PowerBuilder. Αυτό οφείλεται στο γεγονός ότι τυπικά χρησιμοποιείτε μη συμβατικά στοιχεία για εφαρμογές που εκτελούνται σε διακομιστή.

Βιβλιοθήκες

Σε βιβλιοθήκες του PowerBuilder (αρχεία PBL) αποθηκεύετε αντικείμενα, όπως παράθυρα και μενού. Όταν εκτελείτε μια εφαρμογή, το PowerBuilder ανακτά τα αντικείμενα από τη βιβλιοθήκη. Οι εφαρμογές μπορούν να χρησιμοποιούν όσες βιβλιοθήκες θέλετε. Όταν δημιουργείτε μια εφαρμογή, καθορίζετε ποιες βιβλιοθήκες χρησιμοποιεί.

Έργα

Μπορείτε να δημιουργήσετε αντικείμενα του έργου που δημιουργούν εκτελέσιμες εφαρμογές και στοιχεία που μπορείτε να αναπτύξετε σε ένα διακομιστή, καθώς και αντικείμενα διακομιστή μεσολάβησης που χρησιμοποιείτε σε εφαρμογές EA Server.

ΚΕΦΑΛΑΙΟ 5 – Η ΕΦΑΡΜΟΓΗ

Ο κωδικας στο αρχείο connection.php

```
<?php
session_start();

//Για λόγους ασφάλειας του server δεν έχω συμπληρώσει τα πεδία
$servername = "**.*.*.*.*:3306";
$username = "*****";
$password = "*****";
$dbname = "*****";

// Δημιουργία Σύνδεσης
$conn = new mysqli($servername, $username, $password, $dbname);
mysqli_set_charset($conn,"utf8");
// Έλεγχος Σύνδεσης
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

Ο κώδικας στο αρχείο pass.php

```
<?php
session_start();//ξεκινάμε τα session
include ('connection.php');//εισάγουμε το αρχείο για την σύνδεση στην βάση
if($_SERVER["REQUEST_METHOD"] == "POST") {//συνάρτηση αν
εκτελεστεί το POST
    // username και password στέλνονται από την φόρμα

    $username = mysqli_real_escape_string($conn,$_POST['username']);
    $password = mysqli_real_escape_string($conn,$_POST['password']);

    $sql = "SELECT ideal FROM users WHERE email = '$username' and
password = '$password'";
    //εισαγωγή των στοιχείων που επιστρέφει η sql στην μεταβλητή result
    $result = mysqli_query($conn,$sql);//συνάρτηση εκτέλεσης την εντολής
SQL
    $datas = array();
    if (mysqli_num_rows($result)>0){//εισάγουμε τα στοιχεία από την βάση
σε μεταβλητή τύπου Array
        while($row = mysqli_fetch_assoc($result)){
            $datas[]=$row;
        }
    } else {
```

```

        echo '<span style="color:#ff0000;text-align:center;">Το Όνομα
Χρήστη ή ο Κωδικός σας είναι λάθος.<br></span>';
        echo '<span style="color:#ff0000;text-align:center;">Προσπαθήστε
ξανά!<br><br></span>';
    }
    $result = $conn->query($sql);//συνάρτηση εκτέλεσης την εντολής SQL
    //συνάρτηση που μετράει τις γραμμές που επιστρέφει η εντολή SQL
    $count = mysqli_num_rows($result);

    //Εάν οι μεταβλητές αντιστοιχούν με μεταβλητή στη βάση, τότε πρέπει να
επιστρέψει 1 γραμμή

    if($count == 1 )
        { //εισαγωγή σε μεταβλητές SESSION τις χρήσιμες πληροφορίες
του χρήστη
            $_SESSION['login_user'] = $datas[0]['name']."
".$datas[0]['surname'];
            $_SESSION['login_id'] = $datas[0]['id'];
            $_SESSION['email'] = $datas[0]['email'];
            $hh="location:main.php";//link για ανακατεύθυνση
            header($hh);//ανακατεύθυνση
        }
    }
    $conn->close();//συναρτηση για κλεισιμο της συνδεσης στην βαση
?>
<script>
    // Παιρνει το ID του προειδοποιητικού κειμένου
    var text = document.getElementById("text");

    //Όταν ο χρήστης πιέσει οποιοδήποτε πλήκτρο στο πληκτρολόγιο, εκτελεί τη
λειτουργία
    input.addEventListener("keyup", function(event) {
        // Εάν το "caps lock" είναι ενεργό, εμφανίζει το κείμενο προειδοποίησης για
το πρώτο input
        if (event.getModifierState("CapsLock")) {
            text.style.display = "block";
        } else {
            text.style.display = "none"
        }
    });
</script>
<style>
    @import "style.css"; /* επισυνάψει αρχείο που περιέχει κοινά στοιχεία της
Css*/
</style>

<meta name="viewport" content="width=device-width, initial-scale=1">
<html>

```

```

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Δήμος Παιανίας | Είσοδος </title>
</head>

<body>
  <div class="imgcontainer"><!--Κλάση για το λογότυπο του Δήμου-->
    
  </div>

  <div class="container"><!--Κλάση τύπου container για φόρμα σύνδεσης-->
    <form action = "" method = "post"><!--Δημιουργία φόρμας μεθόδου POST
για να στέλνει τα στοιχεία που πληκτρολογεί ο χρήστης στις μεταβλητές PHP -->
      <input type="text" placeholder="Enter Username" name="username"
required>

      <input type="password" id="myInput" placeholder="Enter Password"
name="password" required>
      <p id="text">Προσοχή! Το πλήκτρο CAPS LOCK είναι ενεργό!</p><!--
Μήνυμα για το πλήκτρο CAPS LOCK-->

      <button type="submit">Σύνδεση</button>
    </form>
  </div>

  <div class="container signin"><!--Κλάση τύπου container για πλήκτρο
ανακατεύθυνσης-->
    <p>Δεν έχεις λογαριασμό; <a href="registration.php">Εγγραφή</a>.</p>
  </div>

  <div class="container signin"><!--Κλάση τύπου container για πλήκτρο
ανακατεύθυνσης-->
    <a href="passwordcheck.php">Ξέχασες τον κωδικό σου;</a>
  </div>

</body>
</html>

```

Ο κώδικας το αρχείο main.php

```

<?php
  session_start();
  include ('connection.php');
  $user_id=$_SESSION['login_id'];

  // Check connection
  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
    $hh="location:pass.php";

```

```

    header($hh);
}

if($_SESSION['login_user']=="){
    $hh="location:pass.php";
    header($hh);
}
mysqli_set_charset($conn,"utf8");
//$sql = "SELECT * from dimoi ORDER BY onoma ASC;" ;
//$result = $conn->query($sql);

$sql2 = "SELECT * from katigories order by katigories_id ASC;" ;
$result2 = $conn->query($sql2);

$sql3 = "SELECT name from users where id='$user_id'" ;
$result3 = $conn->query($sql3);
?>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
    @import "style.css";
</style>
<script src="myscripts.js"></script>
<html>

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>
        Δήμος Παιανίας | Καταχώρηση Βλάβης
    </title>
</head>

<body>

    <div id="mySidenav" class="sidenav">
        <a href="javascript:void(0)" class="closebtn"
onclick="closeNav()">&times;</a>
        <a href="profile.php">Προφίλ</a>
        <a href="records.php">Ιστορικό</a>
        <a href="telephones.php">Χρήσιμα Τηλέφωνα</a>
    </div>
    <span style="font-size:30px;cursor:pointer"
onclick="openNav()">&#9776;</span>

    <div class="topright"><?php
        while($rows = $result3->fetch_assoc())
        {
            $name=$rows['name'];
            echo "$name";
        }
    </div>

```

```

    ?>
</div>

<div class="imgcontainer">
  
</div>

<h2 align="center">Καλώς Ήρθατε</h2>
<h3 align="center">Πλατφόρμα καταχώρησης ζημιών και βλαβών</h3>
<div class="container">
  <form id="dimoi" action="success.php" method="post">
    <select name="katigoria" required>
      <option value="" disabled selected>Επιλέξτε Κατηγορία</option>
      <?
        while($rows = $result2->fetch_assoc())
        {
          $name=$rows['katigoria'];
          $katigoria_id=$rows['katigories_id'];
          echo "<option value= '$katigoria_id'>$name</option>";
        }
      ?>
    </select>
    <br><br>
    <select name="proteraiotita" required>
      <option value="" disabled selected>Επιλέξτε Προτεραιότητα</option>
      <option value="Χαμηλή">Χαμηλή</option>
      <option value="Μεσαία">Μεσαία</option>
      <option value="Υψηλή">Υψηλή</option>
      <option value="Άμεσης Ανάγκης">Άμεσης Ανάγκης</option>
    </select required>
    <br><br>
    <input type="text" placeholder="εισάγετε διεύθυνση" name="dieuthinsi"
required>
    <br><br>
    <input type="text" placeholder="εισάγετε περιγραφή" name="perigrifi"
required>
    <br><br>
    <button type="submit">Καταχώρηση</button>
  </form>
</div>
</body>
</html>

```

Ο Κώδικας στο αρχείο success.php

```

<?php
  session_start();//Για να ξεκινήσουν τα session

```

```

include ('connection.php');//εισαγουμε το αρχαιο για την συνδεση στην βαση

//ελεγει εναν υπαρχει session με ID χρηστη,αλλιως ανακατευθυνεται στην
καρτελα εισοδου
if($_SESSION['login_user']== ""){
    $hh="location:pass.php";
    header($hh);
}

$dimos=$_POST['dimoi'];
$perigrabi=$_POST['perigrabi'];
$dieuthinsi=$_POST['dieuthinsi'];
$katigoria=$_POST['katigoria'];
$proteraiotita=$_POST['proteraiotita'];
$user_id=$_SESSION['login_id'];
$email=$_SESSION['email'];
$sql = "INSERT INTO ergasies
(dimos,perigrabi,dieuthinsi,katigoria,user_id,hm_kataxwrhshs,proteraiotita)
VALUES
('1','$perigrabi','$dieuthinsi','$katigoria','$user_id',CURDATE(),'$proteraiotita')";

if ($conn->query($sql) == TRUE) { //Η συναρτηση ελεγει αν η εντολη SQL
εκτελεστηκε με επιτυχια,και τοτε στελνει με email τα στοιχεια της καταχωρησης
    $last_id = $conn->insert_id;//συναρτηση που επιστρεφει το τελευταιο ID στον
πινακα
        $to = $email;
        $subject = "Καταχώρηση βλάβης";
        $txt = "Η καταχωρηση σας στην οδο: ".$dieuthinsi." και
περιγραφή: ".$perigrabi." και αριθμο καταχώρησης ".$last_id." εγινε με επιτυχία"
;
        $headers = "From: e-dimos";
        mail($to,$subject,$txt,$headers);
    }else { //αν δεν εκτελεσται σωστα η εντολη SQL τοτε κανει ανακατευθυνση σε
αλλη σελιδα
        $hh="location:error.php";
        header($hh);
    }
?>

<meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμοζει την ιστοσελιδα σε ολες τις οθονες -->
<style>
    @import 'style.css';/*εισαγωγη των κοινων στοιχειων css */
    .p{
        colour :green;
    }
</style>
<html>
    <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">

```

```

<title>
  Επιτυχής Καταχώρηση
</title>
</head>

<body>
  <div class="imgcontainer"><!--ενότητα για εικονίδιο επιτυχίας-->
    
    <p style="color:green;">Η καταχώρηση έγινε με επιτυχία!</p>
  </div>

  <div class="imgcontainer"><!--ενοτητα πληκτρο ανακατεύθυνσης-->
    <form method="post" action="main.php"><!--Φόρμα πληκτρο
ανακατεύθυνσης-->
      <button type="submit">Επιστροφή στην αρχική</button>
    </form>
  </div>

</body>
</html>

```

Ο κώδικας στο αρχείο error.php

```

<?php
session_start();
error_reporting(0);
include ('connection.php');

$dimos=$_POST['dimoi'];
$perigrafi=$_POST['perigrafi'];
$dieuthinsi=$_POST['dieuthinsi'];
$katigoria=$_POST['katigoria'];
$proteraiotita=$_POST['proteraiotita'];
$user_id=$_SESSION['login_id'];
$email=$_SESSION['email'];

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
if($_SESSION['login_user']== ""){
  $hh="location:pass.php";
}

```

```

    header($hh);
}

mysqli_set_charset($conn,"utf8");
$sql = "INSERT INTO ergasies
(dimos,perigrافي,dieuthinsi,kategoria,user_id,hm_kataxwrhshs,proteraiotita)
VALUES
('1','$perigrافي','$dieuthinsi','$kategoria','$user_id',CURDATE(),'$proteraiotita')";

if ($conn->query($sql) == fa) {
    $last_id = $conn->insert_id;
    $to = $email;
    $subject = "Καταχώρηση βλάβης";
    $txt = "Η καταχώρηση σας στην οδο: ".$dieuthinsi."<br><br> και
περιγραφή: ". $perigrافي ." <br><br>και αριθμο καταχώρησης ".$last_id ." εγινε με
επιτυχία" ;
    $headers = "From: e-dimos";
    mail($to,$subject,$txt,$headers);
} else {
    $hh="location:error.php";
    header($hh);
}
?>
<html>

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}

.container {
padding: 0px;
}

.imgcontainer {
text-align: center;
margin: 24px 0 12px 0;
}

img.avatar {
width: 30%;
border-radius: 60%;
}

```



```

.container {
  padding: 0px;
}
.p {
  colour :green;
}
</style>
</head>
<body>
  <div class="imgcontainer">
    

    <p style="color:red;">Δυστήχως υπήρξε καποιο πρόβλημα!</p>
    <p style="color:red;">Παρακαλώ προσπαθήστε ξανά!</p>
  </div>

  <div class="imgcontainer">
    <form method="post" action="new.php">
      <button type="submit">Επιστροφή στην αρχική</button>
    </div>
</form>
</body>
</html>

```

Ο κώδικας στο αρχείο passwordchange.php

```

<?php
  session_start();//ξεκινανε τα session
  include ('connection.php');//εισαγουμε το αρχειο για την συνδεση στην βαση

  //ελεγχει εναν υπαρχει session με ID χρηστη,αλλιως ανακατευθυνεται στην
  καρτελα εισοδου
  if($_SESSION['login_user']== ""){
    $hh="location:pass.php";
    header($hh);
  }

  $user_id=$_SESSION['login_id'];
  $email=$_SESSION['email'];

  if($_SERVER["REQUEST_METHOD"] == "POST") {//συναρτηση
  αν εκτελεσται το POST
    $oldpassword=$_POST['oldpassword'];
    $password=$_POST['password'];
    $password2=$_POST['password2'];

```

```

        $sql2="select password from users where
id='$user_id' and password='$oldpassword';";
        $result = mysqli_query($conn,$sql2);//παιρνουμε τον παλιο
κωδικο του χρηστη
        $datas = array();
        if (mysqli_num_rows($result)>0){//εισαγουμε τα στοιχεία που
χρειαζομαστε απο την βαση σε μεταβλητη τυπου Array
            while($row = mysqli_fetch_assoc($result)){
                $datas[]=$row;
            }
        }

        $psw = $datas[0]['password'];
        $sql= "UPDATE users SET password='$password'
WHERE id='$user_id';";

        //Ελεγχουμε αν ο παλιος κωδικος είναι σωστος,αν οι
νεοι κωδικοι είναι ίδιοι μεταξύ τους και στην συνεχεια καταχωρουμε και
στελνουμε email ενημερωσης αλλαγης κωδικου
        if ($psw==$oldpassword) {
            if($password == $password2){
                $conn->query($sql);
                $to = $email;
                $subject = "Αλλαγή Στοιχείων Εισόδου";
                $txt = "Πραγματοποιήθηκε αλλαγή στα στοιχεία
εισόδου.Αν δεν γνωρίζετε οτιδήποτε για αυτό,παρακαλείστε να επικοινωνήσετε
αμέσως με τον διαχειριστή του συστήματος";
                $headers = "From: e-dimos";
                mail($to,$subject,$txt,$headers);

                $hh="location:main.php";
                header($hh);
            }else if($password != $password2) {
                echo "Οι κωδικοί δεν
ταιριάζουν";
            }
            else
            {
                echo "Error: " . $sql .
                $conn->close();
            }
        }else{
            echo "Οι κωδικοί δεν ταιριάζουν";
        }
    }
?>
<style>
    @import 'style.css';/* επισυναψει αρχειο που περιεχει κοινα στοιχεία της Css*/

```

```

#message {
  display:none;
  background: #f1f1f1;
  color: #000;
  position: relative;
  padding: 20px;
  margin-top: 10px;
}

#message p {
  padding: 10px 35px;
  font-size: 18px;
}

/* Προσθέτει ένα πράσινο χρώμα κειμένου και ένα σημάδι ✓ όταν οι
απαιτήσεις είναι σωστές */
.valid {
  color: green;
}

.valid:before {
  position: relative;
  left: -35px;
  content: "✓";
}

/* Προσθέτει ένα κόκκινο χρώμα κειμένου και ένα "x" όταν οι απαιτήσεις είναι
λανθασμένες */
.invalid {
  color: red;
}

.invalid:before {
  position: relative;
  left: -35px;
  content: "✘";
}
</style>
<html>
  <!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
  <meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμोजει την ιστοσελιδα σε ολες τις οθονες -->
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Αλλαγή Κωδικού | Δήμος Παιανίας</title>
  </head>

  <body>
  <h2 align="center">Αλλαγή Κωδικού</h2>

```

```

<div class="container"><!--Κλάση τύπου container για φορμα αλλαγης
κωδικου-->
  <form action = "" method = "post">
    <input type="password" id="myInput" placeholder="Εισάγετε τον
παλιο κωδικό" name="oldpassword" required>
    <input type="password" id="myInput2" placeholder="Εισαγετε νέο
κωδικό" name="password" required>
    <input type="password" id="myInput3" placeholder="Εισάγετε
Ξανά" name="password2" required>
    <p id="text">Προσοχή! Το πλήκτρο CAPS LOCK είναι ενεργό!</p>
    <button type="submit">Καταχώρηση</button>
  </form>

</div>
<div id="message"><!--Κλάση για την εμφανιση των καταλληλων
προυποθεσεων του νεου κωδικου-->
  <h3>Ο Κωδικός σας πρέπει να τήρει τι παρακάτω προυποθέσεις:</h3>
  <p id="letter" class="invalid">Έναν <b>Μικρό </b> Γράμμα</p>
  <p id="capital" class="invalid">Έναν <b>Κεφαλαίο </b> Γράμμα</p>
  <p id="number" class="invalid">Έναν <b>Αριθμό</b></p>
  <p id="length" class="invalid">Το λιγότερο <b>8
χαρακτήρες</b></p>
</div>
</body>
<script>
  var input = document.getElementById("myInput");
  var input2 = document.getElementById("myInput2");
  var input3 = document.getElementById("myInput3");

  // Προειδοποιητικό κείμενο
  var text = document.getElementById("text");

  //Όταν ο χρήστης πιέσει οποιοδήποτε πλήκτρο στο πληκτρολόγιο, εκτελεί τη
λειτουργία
  input.addEventListener("keyup", function(event) {

    // Εάν το "caps lock" είναι ενεργο,εμφανίζει το κείμενο προειδοποίησης για το
πρωτο input
    if (event.getModifierState("CapsLock")) {
      text.style.display = "block";
    } else {
      text.style.display = "none"
    }
  });
  //Όταν ο χρήστης πιέσει οποιοδήποτε πλήκτρο στο πληκτρολόγιο, εκτελεί τη
λειτουργία
  input2.addEventListener("keyup", function(event) {

```

```

// Εάν το "caps lock" είναι ενεργο, εμφανίζει το κείμενο προειδοποίησης για το
δευτερο input
if (event.getModifierState("CapsLock")) {
    text.style.display = "block";
} else {
    text.style.display = "none"
}
});
//Όταν ο χρήστης πιάσει οποιοδήποτε πλήκτρο στο πληκτρολόγιο, εκτελεί τη
λειτουργία
input3.addEventListener("keyup", function(event) {

// Εάν το "caps lock" είναι ενεργο, εμφανίζει το κείμενο προειδοποίησης για το
δευτερο input
if (event.getModifierState("CapsLock")) {
    text.style.display = "block";
} else {
    text.style.display = "none"
}
});

var myInput = document.getElementById("myInput2");
var letter = document.getElementById("letter");
var capital = document.getElementById("capital");
var number = document.getElementById("number");
var length = document.getElementById("length");

//Όταν ο χρήστης πατήσει πάνω στο πεδίο για να βάλει τον κωδικό, εμφανίζει το
παραθυρό με τις απαραίτητες προϋποθέσεις
myInput.onfocus = function() {
    document.getElementById("message").style.display = "block";
}

//Όταν ο χρήστης πατήσει πάνω στο πεδίο για να βάλει τον κωδικό, κρύβει το
παραθυρό με τις απαραίτητες προϋποθέσεις
myInput.onblur = function() {
    document.getElementById("message").style.display = "none";
}

// Όταν ο χρήστης αρχίσει να πληκτρολογεί
myInput.onkeyup = function() {
    // Επικύρωση πεζών γραμμάτων
    var lowerCaseLetters = /[a-z]/g;
    if(myInput.value.match(lowerCaseLetters)) {
        letter.classList.remove("invalid");
        letter.classList.add("valid");
    } else {
        letter.classList.remove("valid");
    }
}

```

```

    letter.classList.add("invalid");
  }
  //Παρακατω οσο ο χρηστης πληκτρολογει,ελεγχει και επικυρωνει τις
  προυποθεσεις
  // Επικύρωση κεφαλαίων γραμμάτων
  var upperCaseLetters = /[A-Z]/g;
  if(myInput.value.match(upperCaseLetters)) {
    capital.classList.remove("invalid");
    capital.classList.add("valid");
  } else {
    capital.classList.remove("valid");
    capital.classList.add("invalid");
  }

  // Επικύρωση αριθμών
  var numbers = /[0-9]/g;
  if(myInput.value.match(numbers)) {
    number.classList.remove("invalid");
    number.classList.add("valid");
  } else {
    number.classList.remove("valid");
    number.classList.add("invalid");
  }

  // // Επικύρωση απαιρητου μηκους
  if(myInput.value.length >= 8) {
    length.classList.remove("invalid");
    length.classList.add("valid");
  } else {
    length.classList.remove("valid");
    length.classList.add("invalid");
  }
}
}
</script>
</html>

```

Ο κώδικας στο αρχείο passwordcheck.php

```

<?php
session_start();//ξεκινανε τα session
include ('connection.php');//εισαγουμε το αρχειο για την συνδεση στην βαση

if($_SERVER["REQUEST_METHOD"] == "POST") {//συναρτηση που εκτελειτε
αν εκτελεστει το POST
    // το email στελνετε απο την φορμα

```

```

    $email = mysqli_real_escape_string($conn,$_POST['email']);////συνάρτηση
    που ορίζει ειδικούς χαρακτήρες σε μια συμβολοσειρά για χρήση σε μια εντολή
    SQL.
    $sql = "SELECT id,password FROM users WHERE email='$email'";

    $result = mysqli_query($conn,$sql);//συναρτηση εκτελεσης την εντολης
    SQL
    $datas = array();
    if (mysqli_num_rows($result)>0){
        while($row = mysqli_fetch_assoc($result)){//εισαγουμε τα στοιχεία
        που χρειαζομαστε απο την βαση σε μεταβλητη τυπου Array
            $datas[]=$row;
        }
    }else {//μήνυμα λάθος αν δεν υπαρχει στην βαση το email που
    πληκτρολογησε ο χρηστης
        echo '<span style="color:#ff0000;text-align:center;">Το Ονομα
    Χρήστη ή ο Κωδικός σας είναι λάθος.<br></span>';
        echo '<span style="color:#ff0000;text-align:center;">Προσπαθήστε
    ξανά!<br><br></span>';
    }
    $result = $conn->query($sql);
    //συναρτηση που μετραει το πληθος των γραμμων που επιστρεφει η βαση.
    $count = mysqli_num_rows($result);

    //Εαν το αποτελεσμα απο το username και password ειναι σωστο,το πληθος
    των γραμμών που επιστρέφει πρέπει να ειναι 1
    //επειτα βαζει τυχαιο κωδικο και τον στελνει με email στον χρηστη
    if($count == 1 )
        {
            $_SESSION['login_id'] = $datas[0]['id'];
            $psw=$_SESSION['login_id'];
            $token = bin2hex(random_bytes(3));
            $sql= "UPDATE users SET password='$token' WHERE
    id='$psw'";
            $conn->query($sql);
            $hh="location:pass.php";
            header($hh);
            $to = $email;
            $subject = "Αλλαγή κωδικού";
            $txt = "Ο νεος κωδικός σας ειναι:". $token ;
            $headers = "From: e-dimos";
            mail($to,$subject,$txt,$headers);
        }
    }
    $conn->close();
    session_unset();
    ?>

<html>

```

```

<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
  <meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμόζει την ιστοσελιδα σε ολες τις οθονες -->
  <title>Επαναφορά Κωδικού</title>
</head>

<style>
  @import "style.css";/* επισυναψει αρχειο που περιεχει κοινα στοιχεια της
Css*/
</style>

<body>
  <div class="imgcontainer"><!--Κλάση για το λογότυπο του Δήμου-->
    
  </div>

  <div class="container"><!--Κλάση τύπου container για φορμα ελεγχου
κωδικου-->
    <form action = "" method = "post"><!--Δημιουργια φορμας μεθοδου POST
για να στελνει τα στοιχεια που πληκτρολογει ο χρηστης στις μεταβλητες PHP -->
      <input type="text" placeholder="Εισάγετε το e-mail σας" name="email"
required>
      <button type="submit">Login</button>
    </form>
  </div>

  <div class="container signin"><!--Κλάση τύπου container για πληκτρο
ανακατευθυνσης-->
    <p>Δεν έχεις λογαριασμό; <a href="registration.php">Εγγραφή</a>.</p>
  </div>
</body>
</html>

```

Ο κώδικας στο αρχείο profile.php

```

<?php
  session_start();
  include ('connection.php');
  $user_id=$_SESSION['login_id'];
  //ελεγχει εναν υπαρχει session με ID χρηστη,αλλιως ανακατευθυνεται στην
  καρτελα εισοδου

  if($_SESSION['login_user']=="){
    $hh="location:pass.php";
    header($hh);
  }

```



```

    $sql = "SELECT id,name,surname,email,cellphone from users where
id='$user_id'";
    $result = $conn->query($sql);//συναρτηση εκτελεσης την εντολης SQL
    $result2 = $conn->query($sql);//συναρτηση εκτελεσης την εντολης SQL

?>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script>

function openNav() {
    document.getElementById("mySidenav").style.width = "200px";
}

function closeNav() {
    document.getElementById("mySidenav").style.width = "0";
}

</script>
<style>
    .sidenav {
        height: 100%;
        width: 0;
        position: fixed;
        z-index: 1;
        top: 0;
        left: 0;
        background-color: #111;
        overflow-x: hidden;
        transition: 0.5s;
        padding-top: 60px;
    }

    .sidenav a {
        padding: 8px 8px 8px 32px;
        text-decoration: none;
        font-size: 25px;
        color: #818181;
        display: block;
        transition: 0.3s;
    }

    .sidenav a:hover {
        color: #f1f1f1;
    }

    .sidenav .closebtn {

```

```

position: absolute;
top: 0;
right: 25px;
font-size: 36px;
margin-left: 50px;
}
* {
  box-sizing: border-box;
}

table {
  width:100%;
}
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th {
  background-color:#4CAF50;
  color: white;
}
h2 {
  colour:solid blue;
}

}
th, td {
  padding: 15px;
  text-align: left;
}

tr:nth-child(even) {background-color: #f2f2f2;} /*Εναλλαγή χρωματος ανα
γραμμη */
</style>

<!DOCTYPE html>
<!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
<meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμोजει την ιστοσελιδα σε ολες τις οθονες -->
<html>
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Προφίλ</title>
  </head>

  <body>
    <div id="mySidenav" class="sidenav"><!--ενοτητα για το πλαινο παραθυρο--
  >

```

```

    <a href="javascript:void(0)" class="closebtn"
onclick="closeNav()">&times;</a>
    <a href="main.php">Αρχική</a>
    <a href="records.php">Καταχωρήσεις</a>
    <a href="telephones.php">Χρήσιμα Τηλέφωνα</a>
</div>
<span style="font-size:30px;cursor:pointer"
onclick="openNav()">&#9776;</span><!--ομαδοποιηση στοιχειων-->

<h2 align="center" ><!--Τιτλος με την χρηση της PhP-->
<?php
    while ($row = $result->fetch_assoc()) {
        echo $row['name']. " ". $row["surname"];
    }
?>

<table>
<tbody id="myTable">
<?php
    if ($result2->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
        // output data of each row
        while($row = $result2->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεία απο την βαση
            echo "
                <tr>
                    <tr>
                        <th>Κωδικός Χρήστη </th>
                        <td>". $row["id"]."</td>
                    </tr>
                    <tr>
                        <th>e-mail </th>
                        <td>". $row["email"]."</td>
                    </tr>
                    <tr>
                        <th>Κινητό</th>
                        <td>". $row["cellphone"]."</td>
                    </tr>

                </tr>";
            }
        }else{
            echo "Κανένα Αποτέλεσμα";
        }

        echo "</table>";
        $conn->close();

?>

```

```
<div class="container signin"><!--ενοτητα για ανακατευθυνση-->
  <a href="passwordchange.php">Αλλαγή Κωδικού</a>
</div>
</tbody>
</table>
</body>
</html>
```

Ο κώδικας στο αρχείο records.php

```

<?php
    session_start();//Για να ξεκινήσουν τα session
    include ('connection.php');//εισαγουμε το αρχείο για την σύνδεση στην βάση

    //ελεγχει εαν υπάρχει session με ID χρηστη,αλλιως ανακατευθυνεται στην
    καρτελα εισοδου
    if($_SESSION['login_user']=="){
        $hh="location:pass.php";
        header($hh);
    }

    $user_id=$_SESSION['login_id'];
    $sql = "SELECT id,perigrafi,dieuthinsi,katastasi from ergasies where
    user_id='$user_id' " ;
    $result = $conn->query($sql);//συναρτηση που εκτελει την εντολη SQL
?>
<script>
    function openNav() {//συναρτηση που ανοιγει το πλαινο παραθυρο
        document.getElementById("mySidenav").style.width = "200px";
    }

    function closeNav() {//συναρτηση που κλεινει το πλαινο παραθυρο
        document.getElementById("mySidenav").style.width = "0";
    }
</script>
<!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
<meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμoxζει την ιστοσελιδα σε ολες τις οθονες -->
<style>
    .sidenav {
        height: 100%;
        width: 0;
        position: fixed;
        z-index: 1;
        top: 0;
        left: 0;
        background-color: #111;
        overflow-x: hidden;
        transition: 0.5s;
        padding-top: 60px;
    }

    .sidenav a {
        padding: 8px 8px 8px 32px;
        text-decoration: none;
        font-size: 25px;
        color: #818181;

```

```
display: block;
transition: 0.3s;
}

.sidenav a:hover {
color: #f1f1f1;
}

.sidenav .closebtn {
position: absolute;
top: 0;
right: 25px;
font-size: 36px;
margin-left: 50px;
}

* {
box-sizing: border-box;
}

table {
width:100%;
}
table, th, td {
border: 1px solid black;
border-collapse: collapse;
}
th{
background-color:#4CAF50;
color: white;
}
h2{
colour:solid blue;
}

}
th, td {
padding: 15px;
text-align: left;
}

tr:nth-child(even) {background-color: #f2f2f2;}

</style>
<html>
<body>
```

```

<div id="mySidenav" class="sidenav"><!--ενοτητα για το πλαινο παραθυρο-->
<a href="javascript:void(0)" class="closebtn"
onclick="closeNav()">&times;</a>
<a href="main.php">Αρχική</a>
<a href="profile.php">Προφιλ</a>
<a href="telephones.php">Χρήσιμα Τηλέφωνα</a>
</div>
<span style="font-size:30px;cursor:pointer"
onlick="openNav()">&#9776;</span>

<h3 align="center" >Ιστορικό</h3>

<table>
<thead>
<tr><!--οριζει τις επικεφαλίδες του πίνακα-->
<th>Κωδικός</th>
<th>Περιγραφή</th>
<th>Διεύθυνση</th>
<th>Κατάσταση</th>
</tr>
</thead>

<tbody id="Table_diplomata">
<?php
if ($result->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
// output data of each row
while($row = $result->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
echo
"<tr>
<tr></tr>
<td>". $row["id"]."</td>
<td>". $row["perigrafi"]."</td>
<td>". $row["dieuthinsi"]."</td>
<td>". $row["katastasi"]."</td>
</tr>";
}
}else {
echo "Κανένα Αποτέλεσμα";
}
echo "</table>";
?>
</tbody>
</table>
</body>
</html>

```

Ο Κώδικας στο αρχείο registration.php

```

<?php
    session_start();//Για να ξεκινήσουν τα session
    include ('connection.php');//εισαγούμε το αρχείο για την σύνδεση στην βάση

if (isset($_POST['create'])){//συναρτηση εαν εκτελεστεί το POST
    //ορίζουμε σε μεταβλητές όλα όσα καταχώρησε ο χρήστης
    $name=$_POST['name'];
    $surname=$_POST['surname'];
    $cellphone=$_POST['cellphone'];
    $email=$_POST['email'];
    $password=$_POST['password'];
    $password2=$_POST['password2'];
    $sqltest="SELECT * FROM users WHERE email='$email'";
    //Ελεγχουμε εαν το email που καταχώρησε υπάρχει ήδη.Εαν ναι,τότε
    εμφανίζει μήνυμα
    $result = $conn->query($sqltest);
    if (mysqli_num_rows($result) > 0) {
        echo "Υπάρχει ήδη αυτό το e-mail";
        echo "Προσπαθήστε ξανά";
    }elseif($password == $password2){//Αν οι κωδικοί είναι ίδιοι και δεν υπάρχει
    το email καταχωρημένο,τότε κάνει εγγραφή
        $sql= "INSERT INTO users (name,surname,email,cellphone,password)
VALUES ('$name','$surname','$email','$cellphone','$password)";
        if ($conn->query($sql) == TRUE) {
            $_SESSION['login_user']=$_POST['email'];
            $hh="location:update.php";
            header($hh);
        }else{
            echo "Error: " . $sql . "<br>" . $conn->error;
            $conn->close();
        }
    }elseif($password != $password2){//Ελεγχει αν οι κωδικοί που
    πληκτρολογήσε ο χρήστης είναι ίδιοι
        echo "Οι κωδικοί δεν ταιριάζουν";
    }
}
?>

<!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
<meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμόζει την ιστοσελίδα σε όλες τις οθόνες -->
<link href="https://fonts.googleapis.com/css?family=Raleway" rel="stylesheet">
<style>
    @import "style.css";
    * {
        box-sizing: border-box;
    }

```



```
body {
  background-color: #f1f1f1;
}

#regForm {
  background-color: #ffffff;
  margin: 40px auto;
  font-family: Raleway;
  padding: 20px;
  width: 100%;
  min-width: 300px;
}

h1 {
  text-align: center;
}

input {
  padding: 10px;
  width: 100%;
  font-size: 17px;
  font-family: Raleway;
  border: 1px solid #aaaaaa;
}

/* Μαρκάρει τα πλαίσια εισαγωγής που εμφανίζουν σφάλμα κατά την
επικύρωση */
input.invalid {
  background-color: #ffdddd;
}

/* Κρυβει ολα τα βηματα της φορμας */
.tab {
  display: none;
}

button {
  background-color: #4CAF50;
  color: #ffffff;
  border: none;
  padding: 10px 20px;
  font-size: 17px;
  font-family: Raleway;
  cursor: pointer;
}

button:hover {
  opacity: 0.8;
}
```

```
}

#prevBtn {
  background-color: #bbbbbb;
}

/* Κάνει κύκλους που υποδεικνύουν τα βήματα της φόρμας */
.step {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbbbbb;
  border: none;
  border-radius: 50%;
  display: inline-block;
  opacity: 0.5;
}

.step.active {
  opacity: 1;
}

/* Σημειώνει τα βήματα που έχουν ολοκληρωθεί και είναι έγκυρα */
.step.finish {
  background-color: #4CAF50;
}

/* Το πλαίσιο μηνύματος εμφανίζεται όταν ο χρήστης κάνει κλικ στο πεδίο
κωδικού πρόσβασης */
#message {
  display: none;
  background: #f1f1f1;
  color: #000;
  position: relative;
  padding: 20px;
  margin-top: 10px;
}

#message p {
  padding: 10px 35px;
  font-size: 18px;
}

/* Προσθέτει ένα πράσινο χρώμα κειμένου και ένα σημάδι ✓ όταν οι
απαιτήσεις είναι σωστές */
.valid {
  color: green;
}
```

```

.valid:before {
  position: relative;
  left: -35px;
  content: "✓";
}

/* Προσθέστε ένα κόκκινο χρώμα κειμένου και ένα ✖ όταν οι απαιτήσεις είναι
λανθασμένες */
.invalid {
  color: red;
}

.invalid:before {
  position: relative;
  left: -35px;
  content: "✖";
}
</style>

<!DOCTYPE html>
<html>
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>
      Εγγραφή | Δήμος Παιανίας
    </title>
  </head>

  <body>
    <!--Κλάση για το λογότυπο του δήμου-->
    <div class="imgcontainer">
      
    </div>

    <form id="regForm" method="post" action=""><!--Φόρμα για εγγραφή νέου
χρήστη-->
      <h1>Εγγραφή:</h1>

      <!-- Ενα "παραθυρο" για κάθε βήμα μεσα στην φόρμα -->
      <div class="tab">Ονοματεπώνυμο:
        <p><input class="form-control" type="text" placeholder="Εισάγετε
Όνομα" oninput="this.className = ""name="name" required></p>
        <p> <input class="form-control" type="text"
placeholder="Εισάγετε Επίθετο" oninput="this.className = ""name="surname"
required></p>
      </div>

      <div class="tab">Στοιχεία επικοινωνίας:

```

```

        <p> <input class="form-control" type="text" oninput="this.className =
        ""name="email" placeholder="Εισάγετε e-mail" required></p>
        <p> <input class="form-control" type="text" oninput="this.className =
        "" name="cellphone" placeholder="Εισάγετε τηλέφωνο" required></p>
    </div>

    <div class="tab">Στοιχεία εισόδου:
        <p> <input class="form-control" type="password" id='psw'
        name="password" oninput="this.className = "" placeholder="Εισάγετε Κωδικό"
        required></p>
        <p> <input class="form-control" oninput="this.className = ""
        type="password" id='psw' name="password2" placeholder="Εισάγετε Κωδικό
        Ξανά" required></p>
    </div>

    <!-- Ενοτητα που δειχνει τα βηματα της φόρμας -->
    <div style="overflow:auto;">
        <div style="float:right;">
            <button type="button" id="prevBtn" onclick="nextPrev(-
            1)">Previous</button>
            <button type="submit" id="nextBtn"
            onclick="nextPrev(1)"name="create">Καταχώρηση</button>
        </div>
    </div>

    <!--Κύκλοι που υποδεικνύουν τα βήματα της φόρμας: -->
    <div style="text-align:center;margin-top:40px;">
        <span class="step"></span>
        <span class="step"></span>
        <span class="step"></span>
        <p>Έχεις ήδη λογαριασμό; <a href="pass.php">Σύνδεση</a>.</p> <!--
    Πλήκτρο για ανακατεύθυνση -->
    </div>

    <!--Κλάση για την εμφάνιση των κατάλληλων προϋποθέσεων του κωδικου--
    >
    <div id="message">
        <h3>Ο Κωδικός σας πρέπει να τήρει τι παρακάτω προϋποθέσεις:</h3>
        <p id="letter" class="invalid">Έναν <b>Μικρό </b> Γράμμα</p>
        <p id="capital" class="invalid">Έναν <b>Κεφαλαίο </b> Γράμμα</p>
        <p id="number" class="invalid">Έναν <b>Αριθμό</b></p>
        <p id="length" class="invalid">Το λιγότερο <b>8 χαρακτήρες</b></p>
    </div>
</form>

<script>
    var currentTab = 0; // Ορίζει το το πρωτο παράθυρο πρεπει να είναι το 0
    showTab(currentTab); // εμφανίζει το τωρινό παράθυρο

```

```

function showTab(n) {
  // Αυτή η λειτουργία θα εμφανίσει την καθορισμένη καρτέλα της φόρμας
  ...
  var x = document.getElementsByClassName("tab");
  x[n].style.display = "block";
  //διορθώνει τα κουμπιά Προηγούμενο / Επόμενο:
  if (n == 0) {
    document.getElementById("prevBtn").style.display = "none";
  } else {
    document.getElementById("prevBtn").style.display = "inline";
  }
  if (n == (x.length - 1)) {
    document.getElementById("nextBtn").innerHTML = "Submit";
  } else {
    document.getElementById("nextBtn").innerHTML = "Next";
  }
  //λειτουργία που θα εμφανίσει τη σωστή ένδειξη βημάτων:
  fixStepIndicator(n)
}

function nextPrev(n) {
  // Αυτή η λειτουργία θα καθορίσει ποια καρτέλα θα εμφανιστεί
  var x = document.getElementsByClassName("tab");
  // Έξοδος από τη λειτουργία εάν κάποιο πεδίο στην τρέχουσα καρτέλα
  είναι άκυρο:
  if (n == 1 && !validateForm()) return false;
  // Απόκρυψη της τρέχουσας καρτέλας
  x[currentTab].style.display = "none";
  // Increase or decrease the current tab by 1:
  currentTab = currentTab + n;
  //Όταν φτάσει στο τέλος της φόρμας
  if (currentTab >= x.length) {
    //η φορμα υποβάλλεται
    document.getElementById("regForm").submit();
    return false;
  }
  //εμφανίζει τη σωστή καρτέλα
  showTab(currentTab);
}

function validateForm() {
  //Λειτουργία που ασχολείται με την επικύρωση των πεδίων φόρμας
  var x, y, i, valid = true;
  x = document.getElementsByClassName("tab");
  y = x[currentTab].getElementsByTagName("input");
  // Βρόχος που ελέγχει κάθε πεδίο εισαγωγής στην τρέχουσα καρτέλα
  for (i = 0; i < y.length; i++) {
    // If a field is empty...
    if (y[i].value == "") {

```

```

        // προσθέτει μια κατηγορία "άκυρη" στο πεδίο
        y[i].className += " invalid";
        // ορίζει την τρέχουσα έγκυρη κατάσταση σε λανθασμένη
        valid = false;
    }
}
//Εάν η έγκυρη κατάσταση είναι αληθής, σημειώνει το βήμα ως τελικό και
έγκυρο
if (valid) {
    document.getElementsByClassName("step")[currentTab].className += "
finish";
}
return valid; // επιστρέφει την έγκυρη κατάσταση
}

function fixStepIndicator(n) {
    //Λειτουργία που καταργεί την ενεργή κλάση όλων των βημάτων
    var i, x = document.getElementsByClassName("step");
    for (i = 0; i < x.length; i++) {
        x[i].className = x[i].className.replace(" active", "");
    }
    //... and adds the "active" class on the current step:
    x[n].className += " active";
}

var myInput = document.getElementById("psw");
var letter = document.getElementById("letter");
var capital = document.getElementById("capital");
var number = document.getElementById("number");
var length = document.getElementById("length");

// //Όταν ο χρήστης πιέσει οποιοδήποτε πλήκτρο στο πληκτρολόγιο, εκτελεί
τη λειτουργία
myInput.onfocus = function() {
    document.getElementById("message").style.display = "block";
}

//Όταν ο χρήστης φυγει απο το πεδιο του κωδικου,κρυβει το παραθυρο με
τις απαιτητες προϋποθεσεις
myInput.onblur = function() {
    document.getElementById("message").style.display = "none";
}

// // Όταν ο χρηστης αρχισει να πληκτρολογει
myInput.onkeyup = function() {
    // Validate lowercase letters
    var lowerCaseLetters = /[a-z]/g;
    if(myInput.value.match(lowerCaseLetters)) {

```

```

letter.classList.remove("invalid");
letter.classList.add("valid");
} else {
letter.classList.remove("valid");
letter.classList.add("invalid");
}

//Παρακατω οσο ο χρηστης πληκτρολογει,ελεγχει και επικυρωνει τις
προυποθεσεις
// Επικύρωση κεφαλαίων γραμμάτων
var upperCaseLetters = /[A-Z]/g;
if(myInput.value.match(upperCaseLetters)) {
capital.classList.remove("invalid");
capital.classList.add("valid");
} else {
capital.classList.remove("valid");
capital.classList.add("invalid");
}

//Επικύρωση αριθμών
var numbers = /[0-9]/g;
if(myInput.value.match(numbers)) {
number.classList.remove("invalid");
number.classList.add("valid");
} else {
number.classList.remove("valid");
number.classList.add("invalid");
}

// Επικύρωση απαιρητου μηκους
if(myInput.value.length >= 8) {
length.classList.remove("invalid");
length.classList.add("valid");
} else {
length.classList.remove("valid");
length.classList.add("invalid");
}
}
}
</script>

</body>
</html>

```

Ο κώδικας στο αρχείο `telephones.php`

```

<?php
session_start();session_start();//Για να ξεκινήσουν τα session
include ('connection.php');//εισαγουμε το αρχειο για την συνδεση στην βαση

```

```

$sql = "select * from telephones where omada='0' order by title ASC";
$result = $conn->query($sql);

$sql1 = "select * from telephones where omada='2' order by title ASC";
$result1 = $conn->query($sql1);

$sql2 = "select * from telephones where omada='1' order by title ASC";
$result2 = $conn->query($sql2);

$sql3 = "select * from telephones where omada='3' order by title ASC";
$result3 = $conn->query($sql3);

$sql4 = "select * from telephones where omada='4' order by title ASC";
$result4 = $conn->query($sql4);
?>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>

function openNav() {
    document.getElementById("mySidenav").style.width = "200px";
}

function closeNav() {
    document.getElementById("mySidenav").style.width = "0";
}

window.onscroll = function() {scrollFunction()};

function scrollFunction() {
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop >
20) {
        document.getElementById("myBtn").style.display = "block";
    } else {
        document.getElementById("myBtn").style.display = "none";
    }
}

//Συναρτηση για οταν ο χρηστης πατησει το πληκτρο η σελιδα να ανεβει επανω
function topFunction() {
    document.body.scrollTop = 0; // Για το Safari
    document.documentElement.scrollTop = 0; // Για Chrome, Firefox, IE και
Opera
}

</script>

<style>

```



```
.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
  transition: 0.3s;
}

.sidenav a:hover {
  color: #f1f1f1;
}

.sidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

#myBtn {
  display: none;
  position: fixed;
  bottom: 20px;
  right: 30px;
  z-index: 99;
  border: 2px green;
  outline: none;
  background-color: green;
  color: white;
  cursor: pointer;
  padding: 15px; /
  border-radius: 100px;
  font-size: 18px;
}
```

```

}

#myBtn:hover {
  background-color: #4242f5; /* Add a dark-grey background on hover */
}

* {
  box-sizing: border-box;
}

table {
  width:100%;
}
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
th{
  background-color: #4CAF50;
  color: white;
}
h2{
  colour:solid blue;
}

}
th, td {
  padding: 15px;
  text-align: left;
}

tr:nth-child(even) {background-color: #f2f2f2;}
</style>

<!--Μέθοδος που καθορίζει την κωδικοποίηση χαρακτήρων-->
<meta name="viewport" content="width=device-width, initial-scale=1"><!--
Μεθοδος που προσαρμोजει την ιστοσελιδα σε ολες τις οθονες -->

<!DOCTYPE html>
<html>
  <head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>
      Δήμος Παιανίας
    </title>
  </head>

```

```

<body>

    <button onclick="topFunction()" id="myBtn" title="Go to
top">Top</button><!--Πληκτρο για να ανεβει επανω η σελιδα-->

    <!--ενοτητα για το πλαινο παραθυρο-->
    <div id="mySidenav" class="sidenav">
        <a href="javascript:void(0)" class="closebtn"
onclick="closeNav()">&times;</a>
        <a href="main.php">Αρχική</a>
        <a href="profile.php">Προφιλ</a>
    </div>

    <span style="font-size:30px;cursor:pointer"
onclick="openNav()">#9776;</span>

    <h2 align="center" >Δήμος Παιανίας</h2>

    <h4 align="center" >Άμεσης Ανάγκης</h4>

    <table>
        <thead>
            <tr><!--οριζει τις επικεφαλίδες του πίνακα-->
                <th>Τίτλος</th>
                <th>Τηλέφωνο</th>
                <th>Διεύθυνση</th>
            </tr>
        </thead>
        <tbody id="Table_anagkis">
            <?php
                if ($result->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
                    // Εξοδος δεδομενων καθε γραμμής
                    while($row = $result->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
                        echo "<tr>
                            <tr></tr>
                            <td>". $row["title"]."</td>
                            <td><a href='tel:". $row["phones"]." '>" . $row["phones"].
"</a></td>
                                <td> <a href='http://maps.google.com/?q=".
$row["dieuthinsi"]." '>" . $row["dieuthinsi"]. " </td>
                            </tr>";
                        }
                    }else {
                        echo "0 results";
                    }
                }
                echo "</table>";
            ?>

```

```

</tbody>
</table>

<h4 align="center" >Σχολεία</h4>
<table>
  <thead>
    <tr><!--ορίζει τις επικεφαλίδες του πίνακα-->
      <th>Τίτλος</th>
      <th>Τηλέφωνο</th>
      <th>Διεύθυνση</th>
    </tr>
  </thead>
  <tbody id="Table_sxoleia">
    <?php
      if ($result2->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
        // Εξοδος δεδομενων καθε γραμμής
        while($row = $result2->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
          echo "<tr>
            <tr></tr>
            <td>". $row["title"]."</td>
            <td><a href='tel:". $row["phones"]." '>" . $row["phones"].
"</a></td>
            <td> <a href='http://maps.google.com/?q="
$row["diethinsi"]." '>" . $row["diethinsi"]. " "</td>
            </tr>";
          }
        }else {
          echo "0 results";
        }
        echo "</table>";
      ?>
    </tbody>
  </table>

<h4 align="center" >Δήμος Παιανίας</h4>
<table>
  <thead>
    <tr><!--ορίζει τις επικεφαλίδες του πίνακα-->
      <th>Τίτλος</th>
      <th>Τηλέφωνο</th>
      <th>Διεύθυνση</th>
    </tr>
  </thead>
  <tbody id="Table_dimos">
    <?php

```

```

        if ($result1->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
        // Εξοδος δεδομενων καθε γραμμής
        while($row = $result1->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
            echo "<tr>
                <tr></tr>
                <td>". $row["title"]."</td>
                <td><a href='tel:". $row["phones"]." '>" . $row["phones"].
"</a></td>
                <td> <a href='http://maps.google.com/?q=".
$row["dieuthinsi"]." '>" . $row["dieuthinsi"]. " </td>

                </tr>";
            }
        } else {
            echo "0 results";
        }

        echo "</table>";

    ?>
</tbody>
</table>

<h4 align="center" >Τράπεζες</h4>

<table>
<thead>
    <tr><!--οριζει τις επικεφαλίδες του πίνακα-->
        <th>Τίτλος</th>
        <th>Τηλέφωνο</th>
        <th>Διεύθυνση</th>
    </tr>
</thead>
<tbody id="Table_trapezes">
    <?php
        if ($result3->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την εντολη
SQL επιστρεφει τιμες
        // Εξοδος δεδομενων καθε γραμμής
        while($row = $result3->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
            echo "<tr>
                <tr></tr>
                <td>". $row["title"]."</td>
                <td><a href='tel:". $row["phones"]." '>" . $row["phones"].
"</a></td>
                <td> <a href='http://maps.google.com/?q=".
$row["dieuthinsi"]." '>" . $row["dieuthinsi"]. " </td>

```

```

        </tr>";
    }
} else {
    echo "0 results";
}

echo "</table>";

?>
</tbody>
</table>

<h4 align="center" >Ιεροί Ναοί</h4>

<table>
<thead>
<tr><!--ορίζει τις επικεφαλίδες του πίνακα-->
<th>Τίτλος</th>
<th>Τηλέφωνο</th>
<th>Διεύθυνση</th>
</tr>
</thead>
<tbody id="Table_ekklisies">
<?php
    if ($result4->num_rows > 0) {//ελεγχει αν το αποτελεσμα απο την
εντολη SQL επιστρεφει τιμες
        // Εξοδος δεδομενων καθε γραμμής
        while($row = $result4->fetch_assoc()) {//διαδικασια εμφανισης σε
πινακα στοιχεια απο την βαση
            echo "<tr>
                <tr></tr>
                <td>". $row["title"]." </td>
                <td><a href='tel:". $row["phones"]." '>" . $row["phones"].
"</a></td>
                <td> <a href='http://maps.google.com/?q=".
$row["diuthinsi"]." '>" . $row["diuthinsi"]. " </td>
            </tr>";
        }
    } else {
        echo "0 results";
    }
    echo "</table>";
?>
</tbody>
</table>
</body>
</html>

```

Ο κώδικας στο αρχείο update.php

```

<?php
session_start();
include ('connection.php');
mysqli_set_charset($conn,"utf8");

$email=$_SESSION['login_user'];
$sql="SELECT * from users where email='$email'";
$result = mysqli_query($conn,$sql);
    $datas = array();
    if (mysqli_num_rows($result)>0){
        while($row = mysqli_fetch_assoc($result)){
            $datas[]=$row;
        }
    }
    $result = $conn->query($sql);

    $count = mysqli_num_rows($result);

    //If result matched $myusername and $mypassword, table row must be 1 row

    if($count == 1 )
    {
        $_SESSION['login_user'] = $datas[0]['name']." ".$datas[0]['surname'];
        $_SESSION['login_id'] = $datas[0]['id'];
    }

    $hh="location:main.php";
    header($hh);
    $conn->close();

?>

<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script
>
<style>
body {
background-image: url("background.jpg");
}
body {font-family: Arial, Helvetica, sans-serif;}

```

```

form {border: 0px solid #f1f1f1;}

div.container {
  margin: 0px;
  background-color: #ffffff;
  border: 1px solid black;
  opacity: 0.6;
  filter: alpha(opacity=60); /* For IE8 and earlier */
}

div.container p {
  margin: 0%;
  font-weight: bold;
  color: #000000;
}
.center {
  text-align: center;
}

.button {

  padding: 14px 20px;
  margin: 8px 0;
  border: 2px solid green;
  border-radius: 12px;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}
.container {
  background-color: #f1f1f1;
  padding: 10px;
}
</style>
</head>

<body>
<br></br>
<br></br>
  <div class="container" align="center">
    <h1 align="center">Τα στοιχεία σας άλλαξαν με επιτυχία</h1>
    </div>

  <div class="center">

```



```

    <a align="center" href="main.php" class="btn btn-info"
role="button">Συνέχεια</a>
  </div>
</body>
</html>

```

Ο κωδικας στο αρχείο style.css

```

body {font-family: Arial, Helvetica, sans-serif;}
form {border: 0px solid #f1f1f1;}

input[type=text], input[type=password] {
  background-position: 10px 10px;
  background-repeat: no-repeat;
  width: 100%;
  font-size: 16px;
  padding: 12px 20px 12px 40px;

  margin-bottom: 12px;
  border: 2px solid blue;
  border-radius: 12px;
}

button:hover {
  opacity: 0.8;
}

.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

img.avatar {
  width: 60%;
  border-radius: 60%;
}

.container {
  padding: 0px;
}

/* Set a grey background color and center the text of the "sign in" section */
.signin {
  background-color: #f1f1f1;
  text-align: center;
  padding: 10px 10px;
}

```

```
margin: 8px 0;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;
  }
}

#text {display:none;color:red}

.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
  transition: 0.3s;
}

.sidenav a:hover {
  color: #f1f1f1;
}

.sidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

.container {
  text-align: center;
```

```
margin: 24px 0 12px 0;
}

select{

    width: 100%;
}

div.topright {
position: absolute;
top: 8px;
right: 16px;
font-size: 18px;
}

    select {
        appearance: none;
        outline: 0;
        background-image: none;
        width: 100%;
        height: 8%;
        color: black;
        cursor: pointer;
        border: 2px solid blue;
        border-radius: 12px;
    }
    .select {
        position: relative;
        display: block;
        width: 1em;
        height: 2em;
        line-height: 3;
        overflow: hidden;
        border-radius: .25em;
        padding-bottom: 10px;
    }

* {
    box-sizing: border-box;
}

table {
    width: 100%;
}

table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
```

```

th{
  background-color:#4CAF50;
  color: white;
}
h2{
  colour:solid blue;
}

th, td {
  padding: 15px;
  text-align: left;
}

tr:nth-child(even) {background-color: #f2f2f2;} /*Εναλλαγή χρωματος ανα
γραμμη */

```

Ο Κώδικας στο αρχείο myscripts.js

```

function openNav() {
  document.getElementById("mySidenav").style.width = "250px";
}

function closeNav() {
  document.getElementById("mySidenav").style.width = "0";
}

```

Ο ΚΩΔΙΚΑΣ ΤΟΥ POWERBUILDER

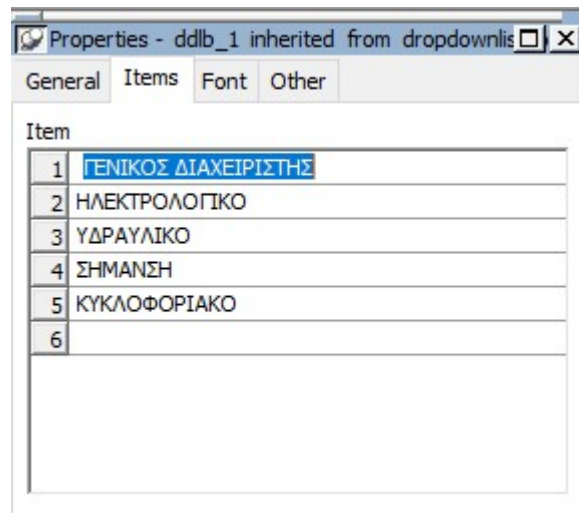
Η εφαρμογή Athens

```
Στο event 'open'  
// Profile Athens  
SQLCA.DBMS = "ODBC"  
SQLCA.AutoCommit = False  
SQLCA.DBParm = "ConnectionString='DSN=web_athens'"  
CONNECT USING SQLCA;  
if SQLCA.SQLCode <> 0 then  
    MessageBox("Προσοχή ", "Δεν έγινε σύνδεση με την Βάση!")  
    Return  
end if  
open (w_pass)
```

Το παράθυρο w_pass



Στο παράθυρο αυτό υπάρχει ένα `drop_down_list_box` που περιέχει τις κατηγορίες που μπορεί να ανοίξει ο χρήστης.



Επιπλέον το πλήκτρο 'ΕΙΣΟΔΟΣ' ελέγχει τι κατηγορία έχει επιλέξει ο χρήστης και ανοίγει το αντίστοιχο παράθυρο.

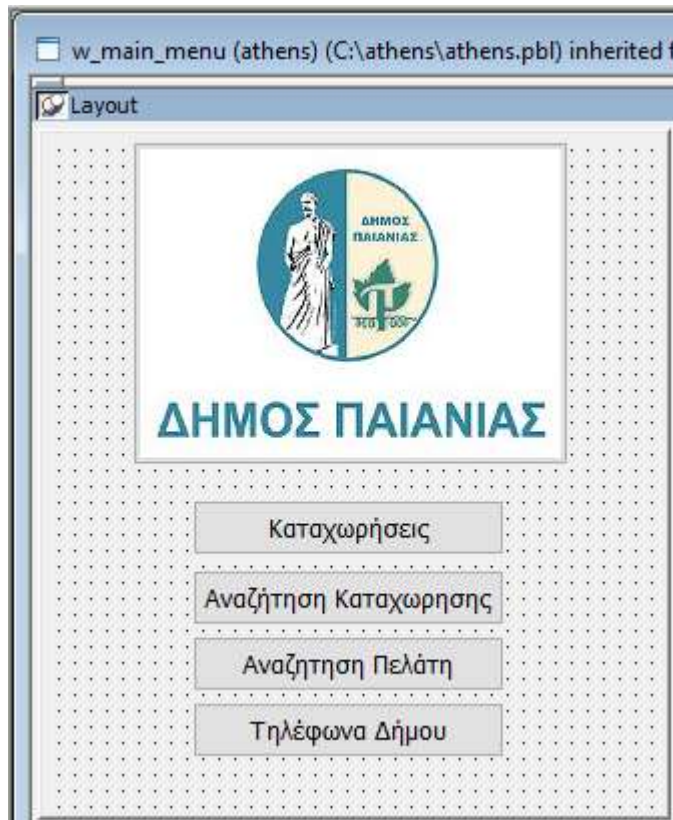
Ο κώδικας στο πλήκτρο 'ΕΙΣΟΔΟΣ'

```

IF (ddlb_1.text="ΗΛΕΚΤΡΟΛΟΓΙΚΟ") THEN
    kwdikos=1
    open (w_ergasies)
ELSEIF (ddlb_1.text="ΥΔΡΑΥΛΙΚΟ") THEN
    kwdikos=2
    open (w_ergasies)
ELSEIF (ddlb_1.text="ΚΥΚΛΟΦΟΡΙΑΚΟ") THEN
    kwdikos=3
    open (w_ergasies)
ELSEIF (ddlb_1.text="ΣΗΜΑΝΣΗ") THEN
    kwdikos=4
    open (w_ergasies)
ELSEIF (ddlb_1.text="ΓΕΝΙΚΟΣ ΔΙΑΧΕΙΡΙΣΤΗΣ") THEN
    open (w_main_menu)
else
    messagebox("Προσοχή","Επιλέξτε κατηγορία")
END IF

```

Το παράθυρο w_main_menu



Το παραθυρο αυτό αποτελεί ένα εργαλείο μόνο για την Γενικό Διαχειριστή του συστήματος. Η χρησιμότητα του είναι να ανοίγει τα παράθυρα που χειρίζεται ο Γενικός Διαχειριστής.

Το Πλήκτρο 'Καταχωρήσεις'

Στο event του πλήκτρου Open υπάρχει ο εξής κώδικας

```
open(w_menu)
```

Το πλήκτρο Αναζήτησης Καταχώρησης

Στο event του πλήκτρου Open υπάρχει ο εξής κώδικας

```
open(w_search)
```

Το πλήκτρο Αναζήτηση Πελάτη

Στο event του πλήκτρου Open υπάρχει ο εξής κώδικας

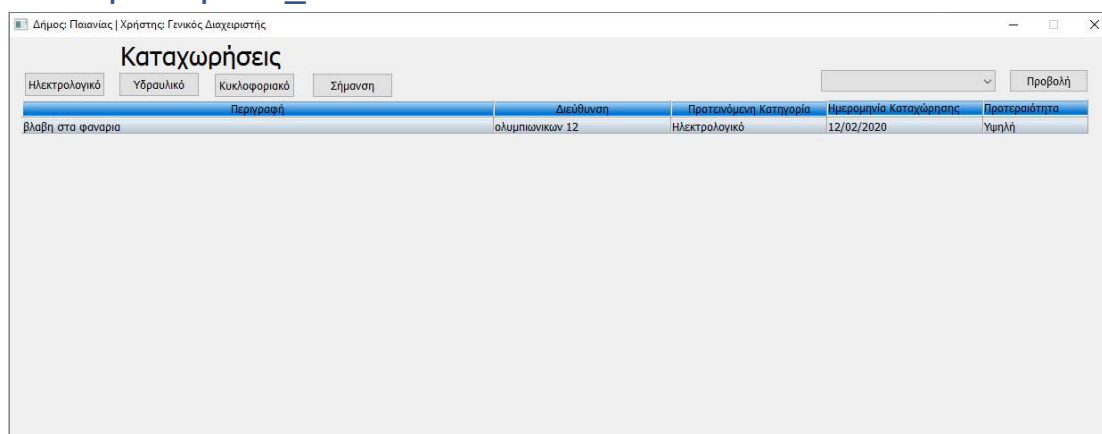
```
open (w_search_users)
```

Το πλήκτρο Τηλεφονά Δήμου

Στο event του πλήκτρου Open υπάρχει ο εξής κώδικας

```
open (w_telephones)
```

Το παραθυρο w_menu



Αποτελεί το βασικό εργαλείο του Γενικού Διαχειριστή. Στο παράθυρο αυτό έχει την δυνατότητα ο διαχειριστής να ελέγχει όλες τις καταχωρήσεις που κάνουν οι πολίτες από την web εφαρμογή και να τις ταξινομεί στις αρμόδιες υπηρεσίες. Πίσω από το παράθυρο αυτό τρέχει μια διαδικασία οπου κρατάει το ιστορικό κίνησης της κάθε καταχώρησης. Επιπλέον έχει την δυνατότητα να δει και να επεξεργαστεί τις ταξινομημένες καταχωρήσεις της κάθε υπηρεσίας(ηλεκτρολογικό, υδραυλικό κ.λπ.). Στο event open του παραθύρου υπάρχει ο εξής κώδικας.

```
DW_1.SETTRANSOBJECT(SQLCA)
dw_1.retrieve()
DW_histo.SETTRANSOBJECT(SQLCA)
dw_histo.retrieve()
```


Το dw_entries

Το data object 'dw_entries' περιέχει το data window 'delegacies' όπου είναι ο πίνακας του γενικού διαχειριστή. Το data window 'delegacies' περιέχει το εξής sq. query:

```
SELECT ergasies. perigrifi,
       ergasies.dieuthinsi,
       ergasies.kategoria,
       ergasies.id,
       ergasies.category,
       ergasies.user_id,
       ergasies.hm_kataxwrhshs,
       ergasies.proteraiotita
FROM ergasies
WHERE ergasies.category = 0
```

Το πλήκτρο 'Ηλεκτρολογικό'

Το πλήκτρο 'Ηλεκτρολογικό' περιέχει τον εξής κώδικα μέσα στον event 'clicked'

```
long cc1,cc2,ld_id
code=dw_entries.GetRow()
cc1 = dw_entries.setItem(code,"category",1)
ld_id=dw_entries.GetItemNumber(code,"id")

cc2=dw_histo.InsertRow(0)
dw_histo.scrolltorow(cc2)
dw_histo.setItem(cc2,"id_ergasias",int(ld_id))
dw_histo.setItem(cc2,"from_node",'0')
dw_histo.setItem(cc2,"to_node",'1')
dw_histo.setItem(cc2,"id_user",43833)
dw_histo.setItem(cc2,"histo_date",today())

dw_histo.update()
dw_histo.retrieve()
dw_entries.update()
dw_entries.retrieve()
```

Το πλήκτρο 'Υδραυλικό'

Το πλήκτρο 'Υδραυλικό' περιέχει τον εξής κώδικα μέσα στον event 'clicked'

```

long cc1,cc2,ld_id
code=dw_entries.GetRow()
cc1 = dw_entries.setItem(code,"category",2)

ld_id=dw_entries.GetItemNumber(code,"id")

cc2=dw_histo.InsertRow(0)
dw_histo.scrolltorow(cc2)
dw_histo.setItem(cc2,"id_ergasias",int(ld_id))
dw_histo.setItem(cc2,"from_node",'0')
dw_histo.setItem(cc2,"to_node",'2')
dw_histo.setItem(cc2,"id_user",43833)
dw_histo.setItem(cc2,"histo_date",today())

dw_histo.update()
dw_histo.retrieve()
dw_entries.update()
dw_entries.retrieve()

```

Το πλήκτρο 'Κυκλοφοριακό'

Το πλήκτρο 'Κυκλοφοριακό' περιέχει τον εξής κώδικα μέσα στον event 'clicked'

```

long cc1,cc2,ld_id
code=dw_entries.GetRow()
cc1 = dw_entries.setItem(code,"category",3)

ld_id=dw_entries.GetItemNumber(code,"id")

cc2=dw_histo.InsertRow(0)
dw_histo.scrolltorow(cc2)
dw_histo.setItem(cc2,"id_ergasias",int(ld_id))
dw_histo.setItem(cc2,"from_node",'0')
dw_histo.setItem(cc2,"to_node",'3')
dw_histo.setItem(cc2,"id_user",43833)
dw_histo.setItem(cc2,"histo_date",today())

dw_histo.update()
dw_histo.retrieve()
dw_entries.update()
dw_entries.retrieve()

```

Το πλήκτρο 'Σήμανση'

Το πλήκτρο 'Σήμανση' περιέχει τον εξής κώδικα μέσα στον event 'clicked'

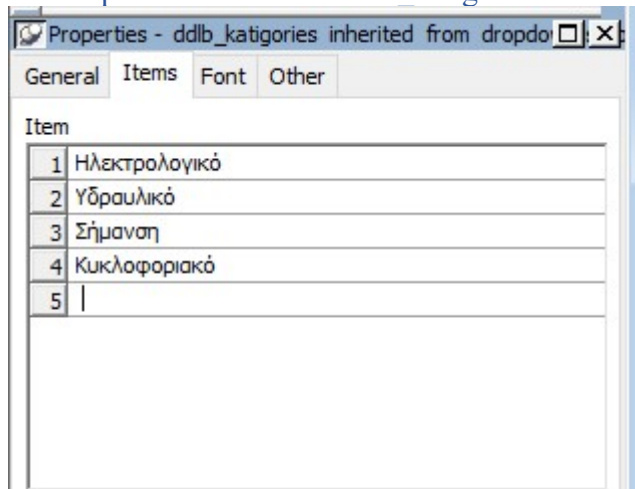
```
long cc1,cc2,ld_id
code=dw_entries.GetRow()
cc1 = dw_entries.setItem(code,"category",4)

ld_id=dw_entries.GetItemNumber(code,"id")

cc2=dw_histo.InsertRow(0)
dw_histo.scrolltorow(cc2)
dw_histo.setItem(cc2,"id_ergasias",int(ld_id))
dw_histo.setItem(cc2,"from_node",'0')
dw_histo.setItem(cc2,"to_node",'4')
dw_histo.setItem(cc2,"id_user",43833)
dw_histo.setItem(cc2,"histo_date",today())

dw_histo.update()
dw_histo.retrieve()
dw_entries.update()
dw_entries.retrieve()
```

Το Drop Down List Box 'ddlb_katigories'



Στο Drop Down List Box 'ddlb_katigories' ο χρήστης επιλέγει την κατηγορία της υπηρεσίας που θέλει να επεξεργαστεί.

Το πλήκτρο 'Προβολή'

Το πλήκτρο 'Προβολή' παίρνει την τιμή που έχει επιλέξει ο χρήστης από το Drop Down List Box 'ddlb_katigories' και ανοίγει το αντίστοιχο παράθυρο. Στο event του πλήκτρου υπάρχει ο παρακάτω κώδικας

```
IF (ddlb_katigories.text="Ηλεκτρολογικό") THEN
    kwdikos=1
    open (w_ergasies)
ELSEIF (ddlb_katigories.text="Υδραυλικό") THEN
    kwdikos=2
    open (w_ergasies)
ELSEIF (ddlb_katigories.text="Κυκλοφοριακό") THEN
    kwdikos=3
    open (w_ergasies)
ELSEIF (ddlb_katigories.text="Σήμανση") THEN
    kwdikos=4
    open (w_ergasies)
ELSE
    messagebox("Προσοχή","Επιλέξτε κατηγορία")
END IF
```

Το dw_histo

Το συγκεκριμένο data object που περιέχει το dw_history δεν είναι εμφανές. Η λειτουργία του είναι να κρατάει το ιστορικό κάθε αιτησης. Το sql query που περιέχει το dw history είναι το εξής:

```
SELECT histo.id,
    histo.id_user,
    histo.from_node,
    histo.to_node,
    histo.id_ergasias,
    histo.histo_date
FROM histo
```

Το παράθυρο w_ergasies

Το παράθυρο w_ergasies είναι το παραθυρο που βλέπουν όλες οι υπηρεσίες(Ηλεκτρολογικό, Υδραυλικό κ.λπ.).Έχει δημιουργηθεί με την κατάλληλη παραμετροποίηση έτσι ώστε να βλέπει η κάθε υπηρεσία τα δικά της δεδομένα.

The screenshot shows a window titled 'Δήμος: Παιανίας | Χρήστης: Γενικός Διαχειριστής | Κατηγορία: Ηλεκτρολογικό'. The window contains a table with the following data:

id	Περιγραφή	Διεύθυνση	Κατάσταση	Προτεραιότητα	Ημ/νία Κατα
77	Βραχυκυκλωμα	Κτήριο 1	Ολοκληρώθηκε	Μεσαία	
80	Βραχυκυκλωμα	Σαλονων 1	Ολοκληρώθηκε	Μεσαία	
121	καμμενη λαμπα	σαλονων 1	Καταχωρήθηκε	Χαμηλή	10/11/2019

Στο event 'close' του w_ergasies υπάρχει ο εξής κώδικας

```
dw_ergasies.update()
```

Στο Event 'open' του w_ergasies το dw_ergasies παίρνει την κατάλληλη τιμή για την επιλεγμένη κατηγορία και γίνεται με τον εξής κώδικα:

```
dw_ergasies.SETTRANSOBJECT(SQLCA)
dw_ergasies.retrieve(kwdikos)
if (kwdikos=1) then
    dw_ergasies.object.title.text="Ηλεκτρολογικό"
    title="Δήμος: Παιανίας | Χρήστης: Γενικός Διαχειριστής | Κατηγορία:
Ηλεκτρολογικό"
elseif (kwdikos=2) then
    dw_ergasies.object.title.text="Υδραυλικό"
    title="Δήμος: Παιανίας | Χρήστης: Γενικός Διαχειριστής | Κατηγορία:
Υδραυλικό"
elseif (kwdikos=3) then
    dw_ergasies.object.title.text="Κυκλοφοριακό"
    title="Δήμος: Παιανίας | Χρήστης: Γενικός Διαχειριστής |
Κατηγορία: Κυκλοφοριακό"
elseif (kwdikos=4) then
    dw_ergasies.object.title.text="Σήμανση"
    title="Δήμος: Παιανίας | Χρήστης: Γενικός Διαχειριστής |
Κατηγορία: Σήμανση"
end if
ddl_filter.text="ΟΛΑ"
dw_history.SETTRANSOBJECT(SQLCA)
dw_history.retrieve()
```

Το Drop Down List Box 'ddlb_filter'

Το Drop Down List Box 'ddlb_filter' περιέχει τον εξής κώδικα μέσα στον event 'itemchanged'

```
string katastasi,DWfilter
IF (ddlb_filter.text="ΕΝΕΡΓΑ") THEN
    DWfilter = "energos='1'"
END IF
IF (ddlb_filter.text="ΟΛΟΚΛΗΡΩΜΕΝΑ") THEN
    DWfilter = "energos='0'"
END IF
IF (ddlb_filter.text="ΥΠΟ ΚΑΤΑΣΚΕΥΗ") THEN
    DWfilter = "energos='2'"
END IF
IF (ddlb_filter.text="ΟΛΑ") THEN
    DWfilter = ""
END IF
dw_ergasies.SetFilter(DWfilter)
dw_ergasies.Filter()
```

Το dw_ergasies

Το data object 'dw_ergasies' περιέχει το data window 'dw_ergasies' οπού είναι ο πίνακας του της εκάστοτε κατηγορίας. Το data window 'dw_ergasies' περιέχει το εξής sql query:

```
SELECT ergasies.id,
    ergasies.perigrafi,
    ergasies.diethinsi,
    ergasies.category,
    ergasies.user_id,
    ergasies.katastasi,
    ergasies.hm_kataxwrhshs,
    ergasies.hm_oloklirwsis,
    ergasies.proteraiotita,
    ergasies.energos
FROM ergasies
WHERE ( ergasies.dimos = '1' ) AND
    ( ergasies.category = :code ) AND
    ( ergasies.katastasi > '0' )
```

Το πλήκτρο 'cb_initialization'

Το πλήκτρο αυτό επιστρέφει την καταχώρηση στο Γενικό Διαχειριστή. Το πλήκτρο 'Σήμανση' περιέχει τον εξής κώδικα μέσα στον event 'clicked'.

```
long cc1,cc2
code=dw_ergasies.GetRow()
int Id_id
```

```
cc1 = dw_ergasies.setItem(code,"category",0)
//////ld_id=dw_ergasies.GetItemNumber(code,"id")

cc2=dw_history.InsertRow(0)
dw_history.scrolltorow(cc2)
dw_history.setItem(cc2,"id_ergasias",int(ld_id))
dw_history.setItem(cc2,"from_node",string(kwdikos))
dw_history.setItem(cc2,"to_node",'0')
dw_history.setItem(cc2,"id_user",43833)
dw_history.setItem(cc2,"histo_date",today())

dw_ergasies.update()
dw_history.update()
dw_history.retrieve()
dw_ergasies.retrieve(kwdikos)
```

Στο παραθύρο w_ergasies υπάρχει και το dw_histo που αναφέρθηκε στο w_menu με τη ίδια ακριβώς λειτουργία.

Το παράθυρο w_report_ergasias

Πληροφορίες Βλάβης

Πληροφορίες Δημότη

Id	0
Όνοματεπώνυμο	Δημητρόπουλος Τάσος
Τηλέφωνο	6971234567
Email	tdimitro17@gmail.com

Πληροφορίες Βλάβης

Id	77	Ενεργή	Ανενεργή	Ημερομηνία Καταχώρησης	00/00/0000
Κατηγορία	1	Προτεραιότητα	Μεσαία	Ημερομηνία Ολοκλήρωσης	10/11/2019
Κατάσταση	ΟΛΟΚΛΗΡΩΘΗΚΕ				
Διεύθυνση	Κτήριο 1				
Περιγραφή	Βραχυκυκλωμα				

Ιστορικό Εργασίας

Απο	Προς	Ημερομηνία	Σχόλια
ΑΡΧΙΚΟΠΟΙΗΣΗ	ΗΛΕΚΤΡΟΛΟΓΙΚΟ	19/03/2020	
ΑΡΧΙΚΟΠΟΙΗΣΗ	ΚΥΚΛΟΦΟΡΙΑΚΟ	12/03/2020	

Στο event 'open' του window w_report_ergasias υπάρχει ο εξής κώδικας:

```
dw_report.settransobject(sqlca)
dw_report.retrieve(Code_xristi,Code_ergasias,Code_ergasias)
dw_dummy.object.cb_insert.fileName='C:\athens\images\print.png'
dw_dummy.object.cb_delete.visible=false
dw_dummy.object.cb_save.visible=false
dw_dummy.object.cb_print.visible=false
```


Το παράθυρο αυτό περιέχει τις πληροφορίες του δημότη που καταχώρησε την βλάβη, τις πληροφορίες της βλάβης καθώς και το ιστορικό της βλάβης. Το παράθυρο `W_report_ergasias` περιέχει:

- ❖ ένα composite data window με όνομα `'dw_report'`
- ❖ ένα datawindow που λειτουργεί σαν sidebar menu με όνομα `'dw_dummy'`

To `dw_report`

Το data object `'dw_report'` περιέχει τα εξής data windows:

- ❖ `dw_users`

Το συγκεκριμένο data window περιέχει τα στοιχεία του χρήστη που έκανε την αίτηση και περιέχει το εξής sql query:

```
SELECT users.id,
       users.name,
       users.surname,
       users.email,
       users.cellphone
FROM users
WHERE users.id = :code
```

Η μεταβλητή `:code` είναι retrieval argument και παίρνει τιμή κατά το άνοιγμα του παραθύρου `'w_report_ergasias'`

- ❖ `dw_profile_ergasias`

Το συγκεκριμένο data window περιέχει τις πληροφορίες της αίτησης και περιέχει το εξής sql query:

```
SELECT ergasies.id,
       ergasies.perigrafi,
       ergasies.dieuthinsi,
       ergasies.katastasi,
       ergasies.category,
       ergasies.user_id,
       ergasies.hm_kataxwrhshs,
       ergasies.hm_oloklirwsis,
       ergasies.energoss,
       ergasies.proteraiotita
FROM ergasies
WHERE ergasies.id = :profile_code
```

Η μεταβλητή `:profile_code` είναι retrieval argument και παίρνει τιμή κατά το άνοιγμα του `w_report_ergasias`

`dw_history_2`

Το συγκεκριμένο data window περιέχει το ιστορικό της αίτησης και περιέχει το εξής sql query:

```
SELECT histo.id,
       histo.id_ergasias,
       histo.id_user,
       histo.from_node,
```

```

        histo.to_node,
        histo.histo_date,
        histo.comment
    FROM histo
    WHERE id_ergasias = :code

```

Η μεταβλητή :profile_code είναι retrieval argument και παίρνει τιμή κατά το άνοιγμα του w_report_ergasias

To dw_dummy

Το dw_dummy έχει ρολό ενός sidebar menu. Αυτό γίνεται διότι ο PowerBuilder δεν μας δίνει την δυνατότητα να έχουμε ένα διαφορετικό menu σε κάθε window με εύκολο τρόπο. Συνεπώς αυτό είναι μια απλή λύση που επιτυγχάνει το σκοπό μας.

Στο event 'clicked' του data object 'dw_dummy' υπάρχει ο εξής κώδικας:

```

if dwo.name='cb_insert' then
    PrintSetup()
    DW_report.Print()
end if

```

Το παραθυρο w_search

The screenshot shows a window titled 'Αναζήτηση' (Search) with a search input field and a 'Προβολή' (View) button. Below is a table with the following data:

Id	Περιγραφή	Διεύθυνση	Κατηγορία	Κατάσταση	Προτεραιότητα	Ημερομηνία
21	εχει σπασει μια λαμπα	σαλωνων 1	Υδραυλικό	registered	Μεσαία	
77	βραχυκυκλωμα	κτήριο 1	Ηλεκτρολογικό	Ολοκληρώθηκε	Μεσαία	
78	γαλασμενο οδαστρωμα	βιτηνης 17		Ολοκληρώθηκε	Υψηλή	
80	βραχυκυκλωμα	Σαλωνων 1	Ηλεκτρολογικό	Ολοκληρώθηκε	Μεσαία	
81	βλαβη στα φαναρια	ολυμπικωνων 12	Κυκλοφοριακό	registered	Υψηλή	12/02/20
120	ολοκληρωθηκε		Ηλεκτρολογικό	Ολοκληρώθηκε		
121	καμμενη λαμπα	σαλωνων 1	Ηλεκτρολογικό	registered	Χαμηλή	10/11/20

Η λειτουργία του παραθύρου w_search είναι να κάνει αναζήτηση μια καταχώρησης με διαφορά κρητηρια. Το παράθυρο μπορεί να το ανοίξει μονο ο γενικός διαχειριστης. Στο event 'open' του παραθύρου υπάρχει ο εξής κώδικας:

```
DW_1.SETTRANSOBJECT(SQLCA)
dw_1.retrieve()
```

Το πλήκτρο 'Προβολή'

Το πλήκτρο προβολή παίρνει την τιμή από πεδίο που συμπληρώνει ο χρήστης και με βάση το κριτηρίου που έχει επιλέξει φιλτράρει τον πίνακα. Το πλήκτρο περιέχει τον εξής κώδικα:

```
string DWfilter

IF (ddlb_1.text="Κωδικός Καταχώρησης") THEN
    DWfilter = "id= "+sle_1.text+"
END IF
IF (ddlb_1.text="Κωδικός Χρήστη") THEN
    DWfilter = "user_id= "+sle_1.text +""
END IF
dw_1.SetFilter(DWfilter)
dw_1.Filter( )
```

Το dw_anazitisi

Το data object 'dw_anazitisi' περιέχει το data window 'dw_search' όπου είναι ο πίνακας με όλες τις καταχωρήσεις. Το data window 'dw_ergasies' περιέχει το εξής sql query:

```
SELECT ergasies.id,
       ergasies.perigrafi,
       ergasies.dieuthinsi,
       ergasies.category,
       ergasies.user_id,
       ergasies.katastasi,
       ergasies.proteraiotita,
       ergasies.hm_kataxwrhshs,
       ergasies.hm_oloklirwsis,
       ergasies.energoss
FROM ergasies
```

Αν ο χρήστης πατήσει διπλό κλικ πάνω σε μια καταχώρηση τότε το πρόγραμμα ανοίγει το παραθυρο w_report_ergasias. Συνεπώς στο event 'doubleclicked' του data object 'dw_anazitisi' υπάρχει ο εξής κώδικας:

```
long S_row

s_row = dw_anazitisi.GetRow()
Code_ergasias=dw_anazitisi.GetItemNumber(s_row,"id")
Code_xristi=dw_anazitisi.GetItemNumber(s_row,"user_id")
open(w_report_ergasias)
```

Το παραθυρο w_search_users

Αναζήτηση Βάση Χρήστη

Id	Όνομα	Επίθετο	Email	Τηλέφωνο
0	Τάσος	Δημητρόπουλος	tdimitr17@gmail.com	6971234567
32	Τάσος	Δημητροπουλοσ	dimitropoulos@insytech.gr	1234
33	Τάσος	Δημητρόπουλος	cse43833@uniwa.gr	
29	Ιωάννα	Βραχνάκη	ionvrachnaki@gmail.com	
30	spiros	dimitropoulos	sp.dimitr@gmail.com	

Id	Περιγραφή	Διεύθυνση	Κατηγορία	Κατάσταση
21	εχει σπασει μια λαμπα	σαλωνων 1	2	registered
77	Βραχουκυκλωμα	Κτήριο 1	1	Ολοκληρώθηκε
80	Βραχουκυκλωμα	Σαλωνων 1	1	Ολοκληρώθηκε
81	βλαβη στα φαναρια	ολυμπικωνικων 12	3	registered
120	ολοκληρωθηκε		1	Ολοκληρώθηκε
121	καμμενη λαμπα	σαλωνων 1	1	registered

Το παραθυρο w_search_users δείχνει όλες τις καταχωρήσεις βάση του επιλεγμένου χρήστη. Το παραθυρο αυτό περιέχει τα εξής:

- ❖ Το Data object dw_users
- ❖ Το Data Object delegacies

To dw_users

To Data Object 'dw_users' περιέχει το data window 'dw_users2' όπου είναι ο πίνακας με όλους τους χρήστες. Το data window 'dw_users2' περιέχει το εξής sql query:

```
SELECT
    users.cellphone,
    users.email,
    users.surname,
    users.name,
    users.id
FROM users
```

Στο data object 'dw_users' ο Γενικός Διαχειριστής επιλέγει τον δημοτή που επιθυμεί και στο data object 'dw_ergasies' εμφανίζονται όλες οι καταχωρήσεις που έχει κάνει ο επιλεγμένος δημοτής. Αυτό επιτυγχάνεται στο event 'doubleclicked' του data object 'dw_users' και περιλαμβάνεται ο εξής κώδικας:

```
long S_row,kwdikos

s_row = Dw_users.GetRow()
kwdikos=dw_users.GetItemNumber(s_row,"id")
dw_ergasies.retrieve(kwdikos)
```

To dw_ergasies

To Data Object 'dw_ergasies' περιέχει το data window 'dw_ergasies_search' όπου είναι ο πίνακας με όλες τις καταχωρήσεις του επιλεγμένου δημότη. Το data window 'dw_ergasies_search' περιέχει το εξής sql query:

```
SELECT ergasies.id,
    ergasies.perigrafí,
    ergasies.dieuthinsi,
    ergasies.category,
    ergasies.user_id,
    ergasies.katastasi,
    ergasies.hm_kataxwrhshs,
    ergasies.hm_oloklirwsis,
    ergasies.energós,
    ergasies.proteraiotita
FROM ergasies
WHERE ergasies.user_id = :code
```

Η μεταβλητή :code είναι retrieval argument και παίρνει τιμή κατά την επιλογή του δημοτή στο Data Object 'dw_users'.

Επιπλέον ο Γενικός Διαχειριστής μπορεί να ανοίξει την καρτέλα της καταχώρησης με διπλό κλικ σε οποιαδήποτε καταχώρηση. Αυτό γίνεται στο event 'doubleclicked' του data Object 'dw_ergasies' και υπάρχει ο εξής κώδικας

```
long S_row

s_row = Dw_ergasies.GetRow()
Code_ergasias=dw_ergasies.GetItemNumber(s_row,"id")
Code_xristi=dw_ergasies.GetItemNumber(s_row,"user_id")
open(w_report_ergasias)
```

Το παραθυρο 'w_telephones

Το παραθυρο 'w_telephones' δείχνει τα σημαντικότερα τηλέφωνα του Δήμου Παιανίας. Ο Γενικός Διαχειριστής μπορεί να επεξεργαστεί, προσθέσει και να διαγράψει τις καταχωρήσεις. Επιπλέον μπορεί να εκτυπώσει την τον πίνακα με όλα τα τηλέφωνα. Στο event 'open' του παραθύρου 'w_telephones' υπάρχει ο εξής κώδικας:

```
dw_telephones.SETTRANSOBJECT(SQLCA)
dw_telephones.RETRIEVE()
```

Επιπροσθετα το παραθυρο 'w_telephones' περιέχει 2 Data Objects

- ❖ Dw_telephones
- ❖ Dw_dummy

Το Dw_telephones

Το Data Object 'dw_telephones' περιέχει το Data Window 'dw_telephones' το οποίο περιέχει τον πίνακα με όλα τα τηλέφωνα του δήμου. Το data window 'dw_telephones' περιέχει το εξής sql query:

```
SELECT telephones.omada,
       telephones.dieuthinsi,
       telephones.title,
       telephones.dimos_id,
       telephones.phones,
       telephones.phones_id
FROM telephones
ORDER BY telephones.omada ASC
```

Το dw_dummy

Το Data object 'dw_dummy' έχει τις ίδιες λειτουργίες όπως αναφέρθηκε και στο παραθυρο 'w_report_ergasias'. Συνεπώς στο event 'clicked' του data Object 'dw_dummy' υπάρχει ο εξής κώδικας:

```
if dwo.name='cb_save' then
    dw_telephones.update()
end if

if dwo.name='cb_insert' then
    long cc1
    cc1=dw_telephones.InsertRow(0)
    dw_telephones.ScrolltoRow(cc1)
end if

if dwo.name='cb_delete' then
    long cc2
    long net
    Net = MessageBox("ΔΙΑΓΡΑΦΗ", "Να διαγράψω την καταχώρηση ?",
Exclamation!, OKCancel!, 2)
    IF Net = 1 THEN
        cc1=dw_telephones.GetRow()
        dw_telephones.DeleteRow(cc2)
```

```
        dw_telephones.update()
    END IF
    dw_telephones.retrieve()
end if

if dwo.name='cb_print' then
    PrintSetup()
    dw_telephones.Print()
end if
```


Βιβλιογραφία

- ❖ <https://el.wikipedia.org/wiki/PHP>
- ❖ <https://el.wikipedia.org/wiki/HTML>
- ❖ <https://www.w3schools.com/css>
- ❖ <https://www.w3schools.com/sql>
- ❖ <https://www.w3schools.com/php>
- ❖ <https://www.php.net/>
- ❖ <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc37772.1250/html/pbgs/CBBBJIEB.htm>

- ❖ Ανάπτυξη Διαδικτυακών Εφαρμογών , Παναγιώτης Δ. Κεντερλής (Αθήνα,2009)

- ❖ Σχεσιακές Βάσεις Δεδομένων , Χρήστος Σκουρλάς,(1^η Έκδοση,2000)

- ❖ Σχεσιακές Βάσεις Δεδομένων , Ευάγγελος Κεχρής (2^η Έκδοση,2015)

- ❖ Συστήματα Διαχείρισης Βάσεων Δεδομένων 3^η έκδοση Ramakrishnan , Gehrke, Εκδόσεις ΤΖΙΟΛΑ

- ❖ Αντικειμενοστρεφή Συστήματα Διαχείρισης Βάσεων Δεδομένων, Π. Βασιλειάδης, Τ. Σελλής. Εκδόσεις Παπασωτηρίου, Αθήνα, 1997.

- ❖ Ασφάλεια Υπολογιστών Αρχές και Πρακτικές 3^η Αμερικανική Έκδοση ,William Stallings , Lawrie Brown,Εκδόσεις Κλειδάριθμος
- ❖ Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
- ❖ Βάσεις Δεδομένων , Αρχές Τεχνολογίας Λογισμικού Γ',Μιχαηλ Ξένος,Δημ. Χριστοδουλακης,Ελληνικο Ανοικτο Πανεπιστημιο