



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΖΩΓΡΑΦΟΥ 157 73, ΑΘΗΝΑ

ΕΒΓΔ-ΔΙΠΛ-00-06

11 Σεπτεμβρίου 2000

Οργάνωση του Μετεωρολογικού Σταθμού του ΕΜΠ σε βάση
δεδομένων προσβάσιμη από το web (ΤΟΜΟΣ Ι: Κείμενο)

Παναγιώτης Θ. Καβαλαγιός

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: **Τίμος Σελλής**



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΓΝΩΣΕΩΝ
ΚΑΙ ΔΕΔΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ

Η διπλωματική αυτή υλοποιήθηκε στους χώρους του Εργαστηρίου Υδραυλικής, του τομέα Υδατικών Πόρων, Υδραυλικών και Θαλάσσιων Έργων υπό την επίβλεψη του καθηγητή κυρίου Τίμου Σελλή. Είναι μία προσπάθεια αναβάθμισης του υπάρχοντος συστήματος απευθείας μετάδοσης μετεωρολογικών δεδομένων στο διαδίκτυο, που λειτουργεί από τον Σεπτέμβριο του 1994. Τα δεδομένα αυτά υπάρχουν χάρης στην προσπάθεια που κατέβαλαν οι κύριοι Νικόλαος Μαμάσης και Αντώνης Χριστοφίδης με επικεφαλή τον κύριο Δημήτριο Κουτσογιάννη.

Θα ήθελα να ευχαριστήσω την ομάδα του μετεωρολογικού σταθμού που ανέφερα, για την πολύτιμη βοήθεια στην εκπόνηση της διπλωματικής μου εργασίας. Με βοήθησαν πολύ να καταλάβω όλες τα μετεωρολογικές λεπτομέρειες που αφορούν τις χρονοσειρές, τα όργανα και τον σταθμό.

11 Σεπτεμβρίου 2000
Πάνος Καβαλαγιός

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	7
1.1	ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ	7
1.1.1	Εισαγωγή.....	7
1.1.2	Ιστορικό.....	7
1.1.3	Τα συστήματα μέτρησης.....	8
1.1.4	Το σύστημα μεταγωγής δεδομένων	9
1.1.5	Το υπολογιστικό σύστημα	11
1.1.6	Καταχώρηση και επεξεργασία δεδομένων	11
1.1.7	Η πρόσβαση στα δεδομένα	12
1.1.8	Σκοπός διπλωματικής.....	12
1.2	ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ	12
2	ΚΕΦΑΛΑΙΟ 2: ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	14
2.1	ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ	14
2.1.1	Εισαγωγή.....	14
2.1.2	Λεξιλόγιο	14
2.1.3	Μοντέλα συστήματος	14
2.1.3.1	Στάδιο εισαγωγής δεδομένων	15
2.1.3.2	Στάδιο επεξεργασίας δεδομένων	15
2.1.3.3	Στάδιο παρουσίασης δεδομένων.....	15
2.1.3.4	Μοντέλο σταδίων του Αίολου	16
2.1.4	Ορισμός λειτουργικών απαιτήσεων	16
2.1.5	Ορισμός μη λειτουργικών απαιτήσεων	17
2.1.6	Εξέλιξη συστήματος.....	17
2.2	ΤΡΕΧΟΥΣΑ ΤΕΧΝΟΛΟΓΙΚΗ ΚΑΤΑΣΤΑΣΗ.....	18
3	ΚΕΦΑΛΑΙΟ 3: ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ	20
3.1	ΠΕΡΙΓΡΑΦΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	20
3.1.1	Εισαγωγή.....	20
3.1.2	Λογικό Σχήμα	20
3.1.2.1	Σταθμός (Station).....	20
3.1.2.2	Βίντεο (Video)	20
3.1.2.3	Φωτογραφία (Photo).....	21
3.1.2.4	Τύπος Σταθμού (Station Type)	21
3.1.2.5	Αιτία (Reason)	21
3.1.2.6	Μη Διαθεσιμότητα Αισθητήρα (Sensor NotAvail).....	22
3.1.2.7	Αισθητήρας (Sensor)	22
3.1.2.8	Μεταβλητή (Variable)	23
3.1.2.9	Χρονικό Βήμα (Time Step).....	23
3.1.2.10	Είδος Χρονοσειράς (Series Types)	23
3.1.2.11	Χρονοσειρά (Time Series).....	24
3.1.2.12	Πρωτογενή Δεδομένα (Data Raw).....	24
3.1.2.13	Επεξεργασμένα Δεδομένα (Data Proc).....	24
3.1.2.14	Συνθετικά Δεδομένα (Data Synth).....	25
3.1.2.15	Σχήμα Οντοτήτων – Συσχετίσεων	25
3.1.3	Φυσικό σχήμα.....	25
3.1.3.1	Πίνακας Σταθμού (Station Table).....	26
3.1.3.2	Πίνακας Βίντεο (Video Table).....	27
3.1.3.3	Πίνακας Φωτογραφίας (Photo Table).....	27
3.1.3.4	Πίνακας Τύπου Σταθμού (Station_Type Table)	27
3.1.3.5	Πίνακας Αιτίας (Reason Table).....	28
3.1.3.6	Πίνακας Μη Διαθεσιμότητας Αισθητήρα (Sensor_NotAvail Table)	28

3.1.3.7	Πίνακας Αισθητήρα (Sensor Table)	28
3.1.3.8	Πίνακας Μεταβλητής (Variable Table)	29
3.1.3.9	Πίνακας Χρονικού Βήματος (Series_Type Table)	29
3.1.3.10	Πίνακας Τύπου Χρονοσειράς (Series_Type Table).....	29
3.1.3.11	Πίνακας Χρονοσειράς (Time Series Table).....	30
3.1.3.12	Πίνακας Πρωτογενών Δεδομένων (Data Table).....	30
3.1.3.13	Πίνακας Επεξεργασμένων Δεδομένων (Data Table).....	30
3.1.3.14	Πίνακας Σύνθετων Δεδομένων (Data Table).....	31
3.2	ΠΕΡΙΓΡΑΦΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΙΟΛΟΣ.....	31
3.2.1	Εισαγωγή.....	31
3.2.2	Εμφάνιση εγγραφών	31
3.2.3	Εισαγωγή μίας εγγραφής.....	32
3.2.4	Διαγραφή μίας εγγραφής.....	32
3.2.5	Ενημέρωση μίας εγγραφής.....	32
3.2.6	Εισαγωγή δεδομένων από αρχείο	33
3.2.7	Έλεγχος δεδομένων	33
3.2.8	Μετατροπή δεδομένων.....	34
3.2.9	Απευθείας εκτέλεση εντολής SQL	36
3.2.10	Παραγωγή γραφημάτων.....	36
3.2.11	Επιλογές του χρήστη	36
3.2.11.1	Εκτύπωση ή απόκρυψη μηνυμάτων ελέγχου	36
3.2.11.2	Σύνδεση σε διαφορετική βάση ή με διαφορετικό κωδικό.....	36
3.2.11.3	Διάβασμα από εναλλακτικό αρχείο παραμετροποίησης.....	36
3.2.11.4	Αυτόματη αποθήκευση ή απόρριψη των αλλαγών	37
3.2.11.5	Σύντομη τεκμηρίωση προγράμματος.....	37
3.3	ΠΕΡΙΓΡΑΦΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΟΥ WEB INTERFACE	37
3.3.1	Ο καιρός τώρα.....	37
3.3.2	Στατιστικά.....	37
3.3.3	Διαγράμματα.....	37
3.3.3.1	Θερμοκρασία	38
3.3.3.2	Βαρομετρική πίεση	38
3.3.3.3	Σχετική υγρασία.....	38
3.3.3.4	Βροχόπτωση.....	38
3.3.3.5	Ηλιακή ακτινοβολία.....	38
3.3.3.6	Διάχυτη ηλιακή ακτινοβολία	38
3.3.3.7	Ολική ηλιακή ακτινοβολία.....	38
3.3.3.8	Ηλιοφάνεια.....	39
3.3.3.9	Μέση ταχύτητα ανέμου.....	39
3.3.3.10	Ριπή ανέμου	39
3.3.3.11	Διεύθυνση ανέμου.....	39
3.3.4	Ιστορικά δεδομένα.....	39
3.3.4.1	Μεταβλητές.....	39
3.3.4.2	Αισθητήρες	39
3.3.4.3	Μορφή αρχείου	40
3.3.4.4	Άδεια χορηγήσης	40
3.3.4.5	Η αίτηση γίνεται δεκτή	40
3.3.5	Γενικές πληροφορίες.....	40
3.3.6	Σύνδεσμοι.....	41
3.3.7	Φωτογραφίες – Βίντεο.....	41
3.3.8	Απαντήσεις ερωτημάτων.....	41
3.3.9	Επικοινωνία – Ομάδα έργου.....	41
3.3.10	Εξελίξεις.....	41
4	ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ	42
4.1	ΠΛΑΤΦΟΡΜΕΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΑ ΕΡΓΑΛΕΙΑ.....	42
4.1.1	Βάση δεδομένων Μετεό.....	42
4.1.2	Προγράμματα aeolus, dbonline, index.cgi και histdata.cgi.....	42

4.1.3	Ιστοσελίδες	42
4.2	ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ	42
4.2.1	Βάση δεδομένων Μετεό	42
4.2.2	Πρόγραμμα Αίολος	47
4.2.2.1	Γενικές δηλώσεις	47
4.2.2.2	Ρουτίνα χρήσης προγράμματος Usage	48
4.2.2.3	Ρουτίνα εκτύπωσης μηνυμάτων sysmsg	49
4.2.2.4	Ρουτίνα εκτύπωσης σφαλμάτων syserr	49
4.2.2.5	Ρουτίνα προσθήκης μονών εισαγωγικών QuoteStrings	50
4.2.2.6	Ρουτίνα εκτέλεσης ερώτησης RunQuery	51
4.2.2.7	Ρουτίνα διαβάσματος ορισμάτων ReadArguments	51
4.2.2.8	Ρουτίνα διαβάσματος αρχείου παραμετροποίησης ReadConfig	58
4.2.2.9	Ρουτίνα ανάγνωσης αρχείου παραμετροποίησης δεδομένων ReadLoadConfig	61
4.2.2.10	Ρουτίνα επιλογής δεδομένων της βάσης SelectData	67
4.2.2.11	Ρουτίνα μετατροπής πολικών από καρτεσιανές συντεταγμένες AngleFromCartesian	68
4.2.2.12	Ρουτίνα εκτύπωσης στατιστικών PrintStatistics	69
4.2.2.13	Ρουτίνα εισαγωγής δεδομένων InsertData	71
4.2.2.14	Ρουτίνα μετατροπής χρονοσειρών TimeSeriesConvert	72
4.2.2.15	Ρουτίνα παραγωγής γραφημάτων GenerateGraphs	82
4.2.2.16	Ρουτίνα δημιουργίας εντολής MakeQuery	91
4.2.2.17	Κυρίως πρόγραμμα (Main)	99
4.2.3	Πρόγραμμα dbonline	102
4.2.4	CGI Perl scripts	102
4.2.4.1	Επιλογή γλώσσας με το index.cgi	102
4.2.4.2	Επιλογή ιστορικών δεδομένων με histdata.cgi	103
4.2.4.3	Πρώτο βήμα ιστορικών δεδομένων με histdata1.cgi	104
4.2.4.4	Δεύτερο βήμα ιστορικών δεδομένων histdata2.cgi	110
4.2.4.5	Τρίτο βήμα ιστορικών δεδομένων histdata3.cgi	117
4.2.4.6	Τέταρτο βήμα ιστορικών δεδομένων histdata4.cgi	123
4.2.4.7	Πέμπτο βήμα ιστορικών δεδομένων histdata5.cgi	131
5	ΚΕΦΑΛΑΙΟ 5: ΕΛΕΓΧΟΣ	143
5.1	ΜΕΘΟΔΟΛΟΓΙΑ ΕΛΕΓΧΟΥ	143
5.2	ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΕΛΕΓΧΟΥ	143
5.2.1	Βάση δεδομένων Μετεό	143
5.2.2	Πρόγραμμα aeolus	144
5.2.3	Πρόγραμμα dbonline	147
5.2.4	Ιστοσελίδες	152
5.2.4.1	Στατιστικά	153
5.2.4.2	Διαγράμματα	154
5.2.4.3	Ιστορικά δεδομένα	156
5.2.4.4	Γενικές πληροφορίες	169
5.2.4.5	Σύνδεσμοι	170
5.2.4.6	Φωτογραφίες – Βίντεο	170
5.2.4.7	Απαντήσεις ερωτημάτων	171
5.2.4.8	Επικοινωνία – Ομάδα Έργου	172
5.2.4.9	Εξελίξεις	173
6	ΚΕΦΑΛΑΙΟ 6: ΕΠΙΛΟΓΟΣ	175
6.1	ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ	175
6.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	175
7	ΒΙΒΛΙΟΓΡΑΦΙΑ	176

1 ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της Διπλωματικής

1.1.1 Εισαγωγή

Στο χώρο της Πολυτεχνειούπολης Ζωγράφου είναι εγκαταστημένος ένας αυτόματος τηλεμετρικός μετεωρολογικός σταθμός, η λειτουργία του οποίου συμπλήρωσε ήδη την επταετία. Στο σταθμό αυτό έχουν δοκιμαστεί ως τώρα διάφοροι τύποι αισθητήρων, αλλά και τεχνικών μέτρησης, καταχώρησης και μεταγωγής δεδομένων και ενεργειακής τροφοδοσίας. Σημαντικό βάρος έχει δοθεί στην άμεση διαθεσιμότητα και εύκολη πρόσβαση των δεδομένων, επίκαιρων και ιστορικών, σε κάθε ενδιαφερόμενο χρήστη. Για το σκοπό αυτό χρησιμοποιήθηκαν οι δυνατότητες του Διαδικτύου και αναπτύχθηκαν εφαρμογές πρόσβασης μέσω του Παγκόσμιου Ιστού.

1.1.2 Ιστορικό

Στα 200 και παραπάνω χρόνια που πέρασαν από την εποχή που πρωτοξεκίνησαν οι συστηματικές μετεωρολογικές παρατηρήσεις (με θεμελιωτή τον άγγλο χημικό και φυσιοδίφη John Dalton, που από το 1787 ενέγραψε 900 000 μετεωρολογικές παρατηρήσεις), οι ανάγκες για την πρόγνωση του καιρού, αλλά και οι ανάγκες για την κατασκευή σημαντικών υδραυλικών έργων, οδήγησαν στην ίδρυση και συστηματική λειτουργία εκτεταμένων δικτύων μετεωρολογικών και υδρομετρικών σταθμών σε όλο τον κόσμο. Τα τελευταία 30 χρόνια εξελίσσεται μια επαναστατική αλλαγή στην τεχνολογία των υδρολογικών και μετεωρολογικών μετρήσεων. Συγκεκριμένα, η ανάπτυξη της μικροηλεκτρονικής επέτρεψε την αντικατάσταση των μηχανικών μετρητικών συσκευών με *αυτόματους* ηλεκτρονικούς αισθητήρες και των συμβατικών μηχανικών καταγραφικών συστημάτων με ψηφιακούς καταχωρητές δεδομένων. Παράλληλα η ανάπτυξη των τηλεπικοινωνιών στον τομέα της μεταγωγής δεδομένων επέτρεψε την άμεση μεταβίβαση των μετρήσεων σε μεγάλες αποστάσεις, την ώρα που αυτές λαμβάνονται, κάνοντας έτσι εφικτή την *τηλεμετρία*. Ωστόσο, η επανάσταση αυτή βρίσκεται ακόμη στα πρώτα της βήματα, δεδομένου ότι οι περισσότεροι εγκατεστημένοι μετρητικοί σταθμοί διεθνώς εξακολουθούν να λειτουργούν με συμβατικά όργανα και τεχνικές μέτρησης.

Στην Ελλάδα, οι μετεωρολογικές μετρήσεις ξεκινούν νωρίς μετά την ίδρυση του ελληνικού κράτους, το 1839, και γίνονται συστηματικές και συνεχείς στην Αθήνα από το 1860 (είναι αξιοσημείωτο το γεγονός ότι την περίοδο 1884-1890 οι μετρήσεις δημοσιεύονταν στην Εφημερίδα της Κυβερνήσεως). Το 1890 εγκαθίσταται ο μετεωρολογικός σταθμός της Αθήνας στη σημερινή του θέση, στο Λόφο Νυμφών. Ο 20ος αιώνας σηματοδοτεί τη δημιουργία δικτύου μετρητικών σταθμών σε όλη τη χώρα.

Από τις αρχές της δεκαετίας του 1990 ξεκίνησαν στην Ελλάδα διαδικασίες για τη μετάβαση στις νέες τεχνολογίες υδρομετεωρολογικών μετρήσεων. Στα πλαίσια αυτά εγκαταστάθηκε στο χώρο της Πολυτεχνειούπολης Ζωγράφου ένας πιλοτικός αυτόματος τηλεμετρικός μετεωρολογικός σταθμός, η λειτουργία του οποίου συμπλήρωσε ήδη την εξαετία. Εξ αρχής ο σταθμός σχεδιάστηκε σαν σύστημα ριζικά διαφορετικό από τα «παραδοσιακά» ελληνικά, που κατά γενική ομολογία χαρακτηρίζονται από βραδείες διαδικασίες τόσο στην ενημέρωση των αρχείων μετρήσεων, όσο και στην παροχή των δεδομένων στους ενδιαφερόμενους, και συχνά από μειωμένη αξιοπιστία των ίδιων των δεδομένων. Ο σχεδιασμός του σταθμού έγινε αρχικά με γνώμονα τις εκπαιδευτικές και ερευνητικές ανάγκες, ενώ στην πορεία προστέθηκε και το κριτήριο της κοινωνικής ενημέρωσης και ωφελιμότητας. Ο εκπαιδευτικός χαρακτήρας του σταθμού επιβάλλει τη συνύπαρξη διαφορετικών τύπων οργάνων, από τα απλά συμβατικά μέχρι τα πιο σύγχρονα ηλεκτρονικά. Η ερευνητική διάσταση του σταθμού κινείται σε δύο επίπεδα: Στο επίπεδο της τεχνολογίας μετρήσεων επιβάλλει

τη δοκιμαστική εφαρμογή και αξιολόγηση διαφορετικών πρακτικών και τεχνολογιών μέτρησης (τύποι αισθητήρων, ενεργειακή τροφοδοσία) και χειρισμού (καταχώρησης, μεταγωγής) δεδομένων. Στο επίπεδο των μετρούμενων μετεωρολογικών μεταβλητών, επιβάλλει την αξιοπιστία και πληρότητα των μετρήσεων με μείωση του κινδύνου διακοπής τους (π.χ. λόγω βλάβης).

Σημαντικό βάρος δόθηκε στην άμεση διαθεσιμότητα (π.χ. σε λίγα λεπτά) και εύκολη πρόσβαση των δεδομένων, επίκαιρων και ιστορικών, σε κάθε ενδιαφερόμενο χρήστη. Για το σκοπό αυτό χρησιμοποιήθηκαν οι δυνατότητες του Διαδικτύου (Internet) και αναπτύχθηκαν εφαρμογές πρόσβασης μέσω του Παγκόσμιου Ιστού (WWW). Είναι αξιοπερίεργο το γεγονός ότι το όλο σύστημα είχε ευρύτατη απήχηση όχι μόνο στους ειδήμονες αλλά και στο πλατύ κοινό (στην Ελλάδα και το εξωτερικό), όπως αποδεικνύεται από τις πολλές καθημερινές προσβάσεις τόσο στα επίκαιρα, όσο και στα ιστορικά δεδομένα.

Ο Αυτόματος Τηλεμετρικός Μετεωρολογικός Σταθμός της Πολυτεχνειούπολης Ζωγράφου εγκαταστάθηκε στη νοτιανατολική πλευρά της Πολυτεχνειούπολης, στους πρόποδες του Υμηττού, έξω από τα κτιριακά συγκροτήματα (Σχ. 1), ακολουθώντας γενικά τις προδιαγραφές του World Meteorological Organization. Ο σταθμός λειτούργησε για πρώτη φορά στις 30 Σεπτεμβρίου 1993, στέλνοντας με τηλεμετάδοση στο κτίριο Υδραυλικής, ανά δεκάλεπτο, μετρήσεις για έξι μετεωρολογικές μεταβλητές: θερμοκρασία, σχετική υγρασία, διεύθυνση-ταχύτητα-ριπή ανέμου και βροχόπτωση, ενώ αργότερα ξεκίνησε και η μέτρηση της ατμοσφαιρικής πίεσης, της ηλιοφάνειας και της ηλιακής ακτινοβολίας. Για την εξασφάλιση της αδιάκοπης παροχής δεδομένων, χωρίς κενά, προωθείται ήδη η ιδέα της διπλής μέτρησης, δηλαδή της χρήσης δύο αισθητήρων για κάθε μεταβλητή, που όπως αποδείχτηκε είναι η μέθοδος που παρέχει τη μέγιστη δυνατή αξιοπιστία.

Στις παραγράφους που ακολουθούν δίνεται η τεχνική περιγραφή του σταθμού, από το επίπεδο των οργάνων και συσκευών μέχρι το επίπεδο εφαρμογών λογισμικού, και η αξιολόγηση των τεχνικών που χρησιμοποιήθηκαν.



Σχ. 1: Ο χώρος του σταθμού και τα όργανα μέτρησης. Δεξιά είναι ο ιστός με τις συσκευές του σταθμού. Αριστερά είναι ένας μετεωρολογικός κλωβός με συμβατικά όργανα, και πίσω του ένα συμβατικό εξατμισίμετρο. Στη μέση ένας συμβατικός βροχογράφος.

1.1.3 Τα συστήματα μέτρησης

Στο χώρο του σταθμού είναι τοποθετημένα δύο αυτόματα τηλεμετρικά υποσύστημα, καθένα από τα οποία αποτελείται από τους αισθητήρες που πραγματοποιούν τις μετρήσεις, τη μονάδα συλλογής που συγκεντρώνει τις μετρήσεις, το κύκλωμα μετάδοσης και το σύστημα

ενεργειακής τροφοδοσίας της μονάδας συλλογής. Οι κεντρικές μονάδες είναι τοποθετημένες πάνω στον ιστό ενώ πάνω σε βραχίονες είναι τοποθετημένοι οι αισθητήρες. Το πρώτο υποσύστημα (Y1) στέλνει τα δεδομένα με τη χρήση ραδιοζεύξης και τροφοδοτείται ενεργειακά από ηλιακό συλλέκτη, ενώ το δεύτερο (Y2) χρησιμοποιεί γραμμή δεδομένων και τροφοδοτείται από γραμμή ρεύματος και εναλλακτικά από συσσωρευτές. Στον Πίν. 1 παρουσιάζονται τα γενικά χαρακτηριστικά του σταθμού και στον

Πίν. 2 τα χαρακτηριστικά των οργάνων μέτρησης.

Πίν. 1: Γενικά χαρακτηριστικά του τηλεμετρικού σταθμού.

Γεωγραφικές συντεταγμένες	
Μήκος	23°47' 16'' ανατολικό
Πλάτος	37°58' 26'' βόρειο
Υψόμετρο	219 μέτρα από τη μέση στάθμη θάλασσας
Διαστάσεις περιφραξης σταθμού	
Μετρούμενες μεταβλητές	Θερμοκρασία, σχετική υγρασία, διεύθυνση-ταχύτητα-ριπή ανέμου, βροχόπτωση, ατμοσφαιρική πίεση, ηλιοφάνεια, ηλιακή ακτινοβολία
Τύποι μονάδων συλλογής	
	Y1: AANDERAA-Norβηγικής κατασκευής Y2: DELTA-T-Αγγλικής κατασκευής
Μήνες με περιορισμένη διαθεσιμότητα μετρήσεων	
	10/93, 9/95, 10/95, 11/95, (διαθεσιμότητα 48%, 79%, 29% και 25% αντίστοιχα)
Ηλεκτρονικές διευθύνσεις σταθμού	
Σελίδες στα ελληνικά	http://www.hydro.ntua.gr/meteo/gr/
Σελίδες στα αγγλικά	http://www.hydro.ntua.gr/meteo/en/
Ηλεκτρονικό ταχυδρομείο	meteo@hydro.ntua.gr

Στο χώρο του σταθμού είναι εγκαταστημένα και συμβατικά μετεωρολογικά όργανα, τα οποία δεν χρησιμοποιούνται συστηματικά για μετρήσεις, αλλά εξυπηρετούν κυρίως εκπαιδευτικές ανάγκες (επίδειξη στους φοιτητές). Χρησιμοποιούνται ακόμη για λόγους σύγκρισης με τα όργανα νέας τεχνολογίας. Έτσι, στα πλαίσια διπλωματικής εργασίας συγκρίθηκαν οι αντίστοιχες μετρήσεις των συμβατικών οργάνων και των ψηφιακών αισθητήρων, με στόχο τον έλεγχο της βαθμονόμησης των τελευταίων.

1.1.4 Το σύστημα μεταγωγής δεδομένων

Η μονάδα ανάγνωσης αισθητήρων Y1 είναι συνδεδεμένη με πομπό VHF και, κάθε φορά που διαβάζονται οι μετρήσεις των αισθητήρων, αυτές διαβιβάζονται στον πομπό. Το σήμα του πομπού λαμβάνεται από δέκτη VHF τοποθετημένο στην ταράτσα του Εργαστηρίου, και από εκεί μεταβιβάζεται με καλώδιο σε μετατροπέα, ο οποίος μεταβιβάζει τη μέτρηση σε σειριακή θύρα υπολογιστή. Η επικοινωνία είναι μονόδρομη: ο πομπός απλώς εκπέμπει, χωρίς να γνωρίζει τι λαμβάνεται στην άλλη άκρη. Κατά συνέπεια, δεν είναι δυνατός ο έλεγχος σφαλμάτων και έτσι, μερικές φορές οι μετρήσεις λαμβάνονται λανθασμένα. Εξ άλλου, το σύστημα λήψης των δεδομένων δεν μπορεί να αποφασίσει πότε θα ληφθούν τα δεδομένα. Πρέπει λοιπόν να τρέχει διαρκώς πρόγραμμα στον υπολογιστή το οποίο να περιμένει συνεχώς τα δεδομένα που μπορεί να σταλούν στη σειριακή θύρα. Τέλος, δεν υπάρχει τρόπος πρόσβασης σε παλαιότερα δεδομένα. Αν

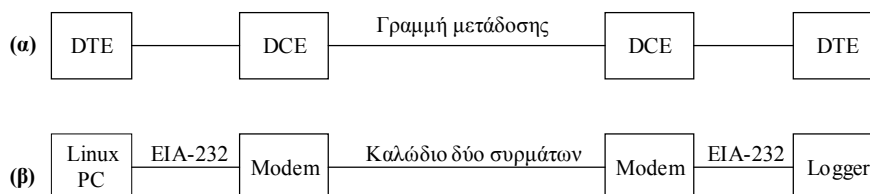
για κάποιο λόγο χαθεί μια μέτρηση, δεν υπάρχει πλέον πρόσβαση σ' αυτήν (παρά μόνο αν διαβαστεί απ' τη μονάδα αποθήκευσης). Το σύστημα λαμβάνει σωστά γύρω στο 95% των μετρήσεων όταν η θερμοκρασία δεν είναι πολύ ψηλή (μέχρι 30°C). Η άνοδος της θερμοκρασίας επηρεάζει αρνητικά την αξιοπιστία της μετάδοσης, με αποτέλεσμα πάνω από τους 35°C η μετάδοση πρακτικά να σταματά.

Πίν. 2: Χαρακτηριστικά των μετρητικών οργάνων.

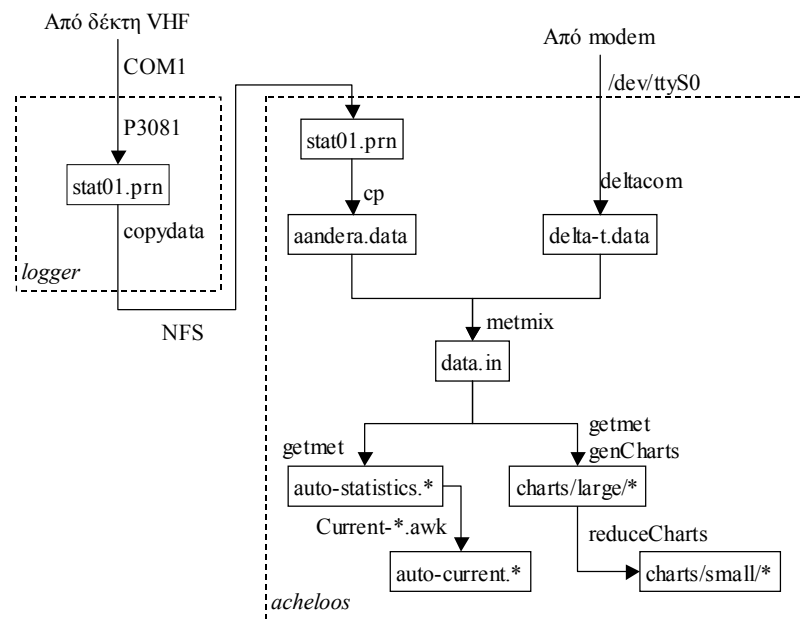
Μεταβλητή	Όργανο*	Κατασκευαστής	Έναρξη μετρήσεων
Ατμοσφαιρική πίεση	Σ-Γ	J. Richard	30/09/1993
	Η-Ψ	Aanderaa	19/02/1994
Βροχόπτωση	Σ-Γ	Lambrecht	30/09/1993
	Η-A	Env. Measurements	16/11/1998
	Η-Ψ	Aanderaa	30/09/1993
Διεύθυνση ανέμου	Η-Ψ	Aanderaa	30/09/1993
Ταχύτητα ανέμου	Η-Ψ	Aanderaa	30/09/1993
Εξάτμιση	Σ	Αναγνώστου	30/09/1993
Ηλιακή ακτινοβολία	Η-A	Kipp& Zonen	16/11/1998
Ηλιοφάνεια	Σ-Γ	Lambrecht	30/09/1993
	Η-Ψ	Aanderaa	19/02/1994
Θερμοκρασία	Σ	Lambrecht	30/09/1993
	Σ-Γ	Lambrecht	30/09/1993
	Η-A	Skye Instruments	10/12/1998
	Η-Ψ	Aanderaa	30/09/1993
Σχετική υγρασία	Σ	Lambrecht	30/09/1993
	Σ-Γ	Lambrecht	30/09/1993
	Η-A	Skye Instruments	10/12/1998
	Η-Ψ	Aanderaa	30/09/1993

* Σ: Συμβατικό όργανο, Σ-Γ: Συμβατικό όργανο με καταγραφικό, Η-Ψ: Ηλεκτρονικός αισθητήρας με ψηφιακό σήμα εξόδου, Η-A: Ηλεκτρονικός αισθητήρας με αναλογικό σήμα εξόδου

Η επικοινωνία του υποσυστήματος Y2 με το Εργαστήριο γίνεται μέσω σειριακών θυρών σε κάθε άκρη. Το μοντέλο της επικοινωνίας φαίνεται στο Σχ. 2 και είναι τυπική περίπτωση. Υπάρχουν δύο τερματικοί σταθμοί (Data Terminal Equipment ή DTE), ο καταχωρητής (logger) και ο υπολογιστής, που χρησιμοποιούν το σύστημα μετάδοσης (το καλώδιο) με μεσολάβηση Data Circuit-terminating Equipment (DCE), που στην περίπτωσή μας είναι τα modem. Η επικοινωνία είναι αμφίδρομη, πράγμα που την απαλλάσσει από σφάλματα αλλά παρουσιάζει και άλλα πλεονεκτήματα. Ο υπολογιστής μπορεί να δώσει διάφορες εντολές στον καταχωρητή, με αποτέλεσμα να είναι δυνατή η πρόσβαση σε όλα τα δεδομένα και η πλήρης διαχείριση του τελευταίου από μακριά.



Σχ. 2: Επικοινωνία μεταξύ σταθμού και εργαστηρίου: (α) θεωρητικό μοντέλο, (β) εφαρμογή στο μετεωρολογικό σταθμό.



Σχ. 3: Λογισμικό που χρησιμοποιείται για τη λήψη και επεξεργασία των δεδομένων.

1.1.5 Το υπολογιστικό σύστημα

Στο Σχ. 3 παρουσιάζεται η πορεία που ακολουθούν τα δεδομένα από τη στιγμή που λαμβάνονται από τους υπολογιστές και η αρχική επεξεργασία που γίνεται σ' αυτά, προκειμένου να είναι έτοιμα για προβολή στο web.

Για την λήψη και επεξεργασία των δεδομένων χρησιμοποιούνται δύο υπολογιστές (Σχ. 3). Ο υπολογιστής logger (PC με λειτουργικό σύστημα MS-DOS) συνδέεται μέσω της σειριακής θύρας με το μετατροπέα και στη συνέχεια το δέκτη VHF του υποσυστήματος Υ1, ενώ λαμβάνει τα δεδομένα τρέχοντας συνεχώς το πρόγραμμα P3081 και τα προωθεί με το πρόγραμμα copydata (DOS TSR – τρέχει στο παρασκήνιο) στον κύριο υπολογιστή Αχελώο (acheloos). Ο Αχελώος είναι σύστημα Linux, και μεταξύ των πολλών λειτουργιών του είναι και web server του Εργαστηρίου. Σ' αυτόν είναι συνδεδεμένο το modem που λαμβάνει τα δεδομένα από το υποσύστημα Υ2, το οποίο οδηγείται από το πρόγραμμα deltacom που τρέχει περιοδικά (ανά 5 λεπτά) μέσω του cron.

Στη συνέχεια υπάρχει μια σειρά προγραμμάτων, όπως φαίνεται στο Σχ. 3, τα οποία διαχειρίζονται τα αρχεία δεδομένων (metmix), παράγουν στατιστικά αποτελέσματα (getmet) και γραφήματα έτοιμα για προβολή στο web (genCharts, reduceCharts). Όλα τα προγράμματα έχουν αναπτυχθεί από την ομάδα λειτουργίας του σταθμού (με εξαίρεση το έτοιμο πρόγραμμα P3081) σε διάφορες γλώσσες προγραμματισμού (Pascal, C, PERL, Unix script).

1.1.6 Καταχώρηση και επεξεργασία δεδομένων

Οι πρωτογενείς μετρήσεις από το υποσύστημα Υ1 αποθηκεύονται ταυτόχρονα στον καταχωρητή και στο αρχείο που συλλέγει τις τηλεμεταδιδόμενες πληροφορίες. Οι τελικές χρονοσειρές των δεκάλεπτων μετρήσεων προκύπτουν κυρίως από τα δεδομένα του καταχωρητή, τα οποία όμως συμπληρώνονται από το αρχείο της τηλεμετάδοσης κατά τα χρονικά διαστήματα όπου ο καταχωρητής έχει απομακρυνθεί από το σταθμό για να αντιγραφούν τα δεδομένα που έχει συλλέξει (περίπου μια φορά κάθε 45 μέρες). Οι πρωτογενείς μετρήσεις από το υποσύστημα Υ2 αποθηκεύονται ταυτόχρονα στη μονάδα και στο αρχείο που συλλέγει τις πληροφορίες μέσω της γραμμής δεδομένων που αποτελούν και τις τελικές χρονοσειρές των ανά δεκάλεπτο μετρήσεων.

Οι τελικές μετρήσεις των μεταβλητών χρησιμοποιούνται για την εξαγωγή χρονοσειρών σε ωριαία, ημερήσια, μηνιαία και ετήσια χρονική κλίμακα, ενώ υπολογίζονται και τα στατιστικά χαρακτηριστικά των χρονοσειρών αυτών. Για την καταχώρηση σε ημερήσια κλίμακα, η ημέρα δεν

έχει οριστεί ημερολογιακά αλλά κατά το ελληνικό (άτυπο) πρότυπο υδρολογικών παρατηρήσεων, δηλαδή η ημερήσια μέτρηση αφορά στο χρονικό διάστημα από τις 8:00 π.μ. της προηγούμενης ημέρας μέχρι τις 8:00 π.μ. της παρούσας ημέρας, στην οποία και αντιστοιχίζεται η παρατηρημένη τιμή. Αντίστοιχα, η καταχώρηση σε ετήσια κλίμακα γίνεται με βάση το υδρολογικό έτος που ξεκινά την 1η Οκτωβρίου και λήγει στις 30 Σεπτεμβρίου. Η εξαγωγή χρονοσειρών πραγματοποιείται σε περιβάλλον Unix και με τη χρήση προγραμμάτων που γράφτηκαν σε γλώσσα C, ενώ η αποθήκευση και η περαιτέρω επεξεργασία των χρονοσειρών γίνονται σε φύλλα εργασίας Excel. Για κάθε τιμή των χρονοσειρών αυτών καταχωρείται επίσης η αντίστοιχη διαθεσιμότητα μετρήσεων εκφρασμένη σε ποσοστό του χρόνου.

1.1.7 Η πρόσβαση στα δεδομένα

Από το πρώτο στάδιο λειτουργίας του σταθμού, τα δεδομένα ήταν διαρκώς προσβάσιμα μέσω του Διαδικτύου, στην αρχή μέσω της υπηρεσίας finger (τότε δεν είχε καθιερωθεί ακόμη ο παγκόσμιος ιστός) και αργότερα μέσω ιστοσελίδων. Η σχετική ιστοσελίδα του Κέντρου Δικτύων του ΕΜΠ πολύ σύντομα έγινε μια από τις δημοφιλέστερες σελίδες του ΕΜΠ (με ρυθμό που ξεπέρασε τη μία πρόσβαση ανά λεπτό). Για ανταπόκριση στο αυξημένο ενδιαφέρον του κοινού δημιουργήθηκαν λεπτομερείς ιστοσελίδες στα ελληνικά και στα αγγλικά με τα επίκαιρα δεδομένα καιρού. Έτσι διατίθενται στις ιστοσελίδες οι πιο πρόσφατες μετρήσεις, τα στατιστικά στοιχεία του τελευταίου εικοσιτετράωρου, καθώς και διαγράμματα χρονικής εξέλιξης των μετεωρολογικών μεταβλητών κατά το τελευταίο εικοσιτετράωρο (βλ. σχετικές ιστοσελίδες στη διεύθυνση του σταθμού που δίνεται στον Πίν. 1).

Εκτός από την προβολή στο web των επίκαιρων δεδομένων, είναι στη διάθεση του ενδιαφερόμενου κοινού το αρχείο των ιστορικών δεδομένων του σταθμού. Ο ενδιαφερόμενος χρήστης μπορεί να λάβει ιστορικά δεδομένα συμπληρώνοντας μια απλή αίτηση (φόρμα) στο web, την οποία το σύστημα επεξεργάζεται αυτόματα και στέλνει αμέσως με e-mail στο χρήστη τα δεδομένα που θέλει. Διατίθενται ωριαία και ημερήσια δεδομένα για όλα τα προηγούμενα υδρολογικά έτη.

1.1.8 Σκοπός διπλωματικής

Καθώς περνούν τα χρόνια, όχι μόνο αυξάνεται ο όγκος των μετεωρολογικών δεδομένων, που σιγά σιγά συσσωρεύονται (αυτή τη στιγμή υπάρχουν περίπου 2 εκατομμύρια μετρήσεις), αλλά αυξάνονται και οι απαιτήσεις των επισκεπτών, που ζητούν πιο αναλυτικά διαγράμματα των μετεωρολογικών μεταβλητών. Επιπλέον, οι υδρομετεωρολόγοι του Εργαστηρίου έχουν αντίστοιχες απαιτήσεις για αποθήκευση των δεδομένων σε διάφορες μορφές (πρωτογενή, διορθωμένα, επεξεργασμένα) και για τη διαχείρισή τους. Τα αρχεία ASCII που αποθηκεύονται τώρα τα δεδομένα, δεν επαρκούν πλέον για να καλύψουν αυτές τις ανάγκες και απαιτείται ανάπτυξη βάσης δεδομένων.

Η διπλωματική αυτή έχει ως αντικείμενο την εισαγωγή της τεχνολογίας των βάσεων δεδομένων, για την διαχείριση και στατιστική επεξεργασία μετεωρολογικών δεδομένων. Εκτός από την βάση δεδομένων που υλοποιήθηκε για να κρατήσει τα δεδομένα των μετεωρολογικών μεταβλητών που δίνουν οι αισθητήρες, δημιουργήθηκε ένα web interface για την παρουσίαση των δεδομένων και πρόγραμμα για την διαχείρισή τους.

1.2 Οργάνωση του τόμου

Ο πρώτος τόμος της διπλωματικής εργασίας χωρίζεται σε 6 συνολικά κεφάλαια. Το κάθε κεφάλαιο αποτελεί μία ενότητα. Στο πρώτο κεφάλαιο γίνεται μία εισαγωγή στο αντικείμενο της διπλωματικής και περιγράφεται η οργάνωση που έχει ο τόμος.

Στο δεύτερο κεφάλαιο περιγράφεται το πρόβλημα. Θα αναλυθούν οι απαιτήσεις των χρηστών και του συστήματος για την συγκεκριμένη υλοποίηση. Επίσης θα γίνει αναφορά στην τρέχουσα τεχνολογική κατάσταση που ισχύει την στιγμή της υλοποίησης.

Η ανάλυση και η σχεδίαση θα παρουσιαστεί στο τρίτο κεφάλαιο. Περιγράφεται η μορφή και η αρχιτεκτονική που έχουν τα εργαλεία που χρησιμοποιήθηκαν και παρουσιάζονται οι ενότητες που τα αποτελούν.

Η υλοποίηση παρουσιάζεται στο κεφάλαιο τέσσερα. Σε αυτό θα δοθούν οι πλατφόρμες ανάπτυξης και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν. Αποκλίσεις από την σχεδίαση θα δοθούν στο ίδιο κεφάλαιο.

Στο κεφάλαιο 5 περιγράφεται ο έλεγχος της ορθής λειτουργίας του συστήματος. Με την μεθοδολογία του μαύρου κουτιού. Πραγματοποιείται μία αναλυτική παρουσίαση του ελέγχου.

Στο τελευταίο κεφάλαιο δίνεται ο επίλογος. Συνοψίζονται τα θέματα που αναλύθηκαν και δίνονται τα συμπεράσματα που προκύπτουν. Τέλος παρουσιάζονται και οι πιθανές μελλοντικές επεκτάσεις του συστήματος.

2 ΚΕΦΑΛΑΙΟ 2: ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

2.1 Απαιτήσεις χρηστών

2.1.1 Εισαγωγή

Ο Τομέας Υδατικών Πόρων, Υδραυλικών και Θαλάσσιων Έργων του Τμήματος Πολιτικών Μηχανικών του ΕΜΠ έχει εγκαταστήσει στο χώρο της Πολυτεχνειούπολης Ζωγράφου ένα μετεωρολογικό σταθμό, που περιλαμβάνει αισθητήρες μετεωρολογικών μεταβλητών. Αυτοί οι αισθητήρες είναι οι εξής:

- Αισθητήρας θερμοκρασίας (2)
- Αισθητήρας υγρασίας (2)
- Αισθητήρας ατμοσφαιρικής πίεσης
- Αισθητήρας διεύθυνσης ανέμου
- Αισθητήρας βροχόπτωσης (2)
- Αισθητήρας ηλιοφάνειας
- Αισθητήρας ταχύτητας ανέμου
- Αισθητήρας ηλιακής ακτινοβολίας

Οι τέσσερις πρώτοι αισθητήρες ανήκουν στην κατηγορία των αισθητήρων στιγμιαίων μεγεθών, οι δύο επόμενοι στην κατηγορία των αισθητήρων αθροιστικών μεγεθών και ο τελευταίος καταγράφει μόνο τη μέγιστη και τη μέση τιμή σε συγκεκριμένο χρονικό διάστημα.

Το γεωγραφικό πλάτος του αυτόματου μετεωρολογικού σταθμού είναι 37°58'26" βόρειο και το γεωγραφικό μήκος είναι 23°47'16" ανατολικό. Το υψόμετρο εδάφους είναι 219 μέτρα από την επιφάνεια της θάλασσας.

Η υπάρχουσα κατάσταση του μετεωρολογικού σταθμού, δίνει τη δυνατότητα παρουσίασης των μετρήσεων των οργάνων από το WWW (με τις τρέχουσες τιμές των οργάνων και διαγράμματα για κάθε μεταβλητή) καθώς και πρόσβαση στο ιστορικό αρχείο του σταθμού και αυτόματη αποστολή των δεδομένων στον χρήστη. Υπάρχει ανάγκη για εκσυγχρονισμό και βελτίωση της υπάρχουσας κατάστασης, επειδή η επεξεργασία και ο χειρισμός των δεδομένων, γίνεται με απλά αρχεία ASCII, που έχουν σαν αποτέλεσμα τη προβολή στατιστικών στοιχείων μόνο για το τελευταίο εικοσιτετράωρο και έναν περιορισμό γενικότερα στην εκμετάλλευση των δεδομένων. Στο έργο αυτό οργανώνονται όλα τα δεδομένα σε μία βάση δεδομένων, με δυνατότητες παρουσίασης στατιστικών και διαγραμμάτων στο παρελθόν. Υποστηρίζονται όλες οι τρέχουσες δυνατότητες του υπάρχοντος συστήματος και φυσικά θα εμπλουτιστεί και με νέες δυνατότητες που θα υπαγορεύσει η ευκολία στον χειρισμό των δεδομένων από την νέα μορφή οργάνωσης.

2.1.2 Λεξιλόγιο

ΣΔΜΔ: Σύστημα Διαχείρισης Μετεωρολογικών Δεδομένων.

Αίολος (ή Aeolus): Το όνομα του ΣΔΜΔ.

Μετεό (ή Meteo): Το όνομα της βάσης δεδομένων

2.1.3 Μοντέλα συστήματος

Το μοντέλο του ΣΔΜΔ Αίολος αποτελείται από τρία βασικά μέρη ή στάδια.

2.1.3.1 Στάδιο εισαγωγής δεδομένων

Το πρώτο στάδιο είναι το στάδιο της εισαγωγής των πρωτογενών ανεπεξέργαστων δεδομένων. Η εισαγωγή αυτή μπορεί να γίνει με τρεις διαφορετικούς τρόπους:

- Με ασύρματη λήψη των δεδομένων από το σταθμό (εκπομπή κάθε 10 λεπτά)
- Με μόνιμη σύνδεση modem (αποστολή κάθε 10 λεπτά)
- Με απευθείας εισαγωγή δεδομένων από το συλλέκτη (μονάδα αποθήκευσης δεδομένων) που έχει ο σταθμός

Στον σταθμό γίνεται διπλή καταγραφή των μετρήσεων που είναι για εκπομπή. Τα δεδομένα καταγράφονται σε ένα συλλέκτη δεδομένων, ο οποίος θα πρέπει να αδειάζει μία φορά το μήνα περίπου. Αυτά είναι πιο έγκυρα από τα δεδομένα που λαμβάνονται ασύρματα, γιατί δεν υπεισέρχονται λάθη μετάδοσης. Η μόνιμη σύνδεση με modem, γίνεται στα καινούργια όργανα με τα οποία εξοπλίστηκε πρόσφατα ο σταθμός και για τα οποία το ΣΔΜΔ Αίολος θα πρέπει να λάβει πρόνοια. Τα δεδομένα που στέλνονται με το modem, καταγράφονται και στον σταθμό, σε μία αποθηκευτική μονάδα που είναι εκεί. Επειδή υπάρχει μεγάλη αξιοπιστία αυτού του τρόπου μετάδοσης, θεωρούμε έγκυρα τα δεδομένα που λαμβάνονται με αυτόν τον τρόπο μετάδοσης. Ο σκοπός είναι να έχουμε διπλούς τους αισθητήρες, για μεγαλύτερη αξιοπιστία στις μετρήσεις. Έτσι τα παλιά όργανα θα λειτουργούν με τα καινούργια παράλληλα, δίνοντας μετρήσεις που θα μπορούν να συσχετιστούν για καλύτερο αποτέλεσμα.

2.1.3.2 Στάδιο επεξεργασίας δεδομένων

Σε αυτό το στάδιο περιλαμβάνονται όλες οι ενέργειες που κάνει ο Αίολος όπως είναι:

- Έλεγχος για λάθη. Σε αυτή την κατηγορία συμπεριλαμβάνεται ο έλεγχος ακραίων τιμών (για παράδειγμα η θερμοκρασία θα πρέπει να είναι μεγαλύτερη από κάποια ελάχιστη τιμή και μικρότερη από κάποια μέγιστη τιμή για την συγκεκριμένη εποχή), ο έλεγχος για χρονική συνέπεια (για παράδειγμα δεν μπορεί η τιμή από δεκάλεπτο σε δεκάλεπτο να έχει πολύ μεγάλη διακύμανση) και ο έλεγχος εσωτερικής συνέπειας (για παράδειγμα δεν μπορεί να βρέχει και να μην έχει σύννεφα)
- Αποθήκευση των δεδομένων στη βάση
- Επεξεργασία και μετατροπή των δεδομένων σε κανονικές δεκάλεπτες, ωριαίες, ημερήσιες και μηνιαίες μετρήσεις
- Παροχή κατάλληλης διεπαφής για αλληλεπίδραση με το χρήστη
- Εξαγωγή δευτερογενών δεδομένων (αποτελέσματα που μπορούν να εξαχθούν με την συμμετοχή των μετρήσεων των οργάνων σε τύπους, για την εύρεση της τιμών άλλων μεταβλητών)

Αυτό είναι από τα σημαντικότερα στάδια της πορείας των δεδομένων και πιθανή δυσλειτουργία και ελαττώματα επηρεάζουν τη συνολική συμπεριφορά του Αίολου.

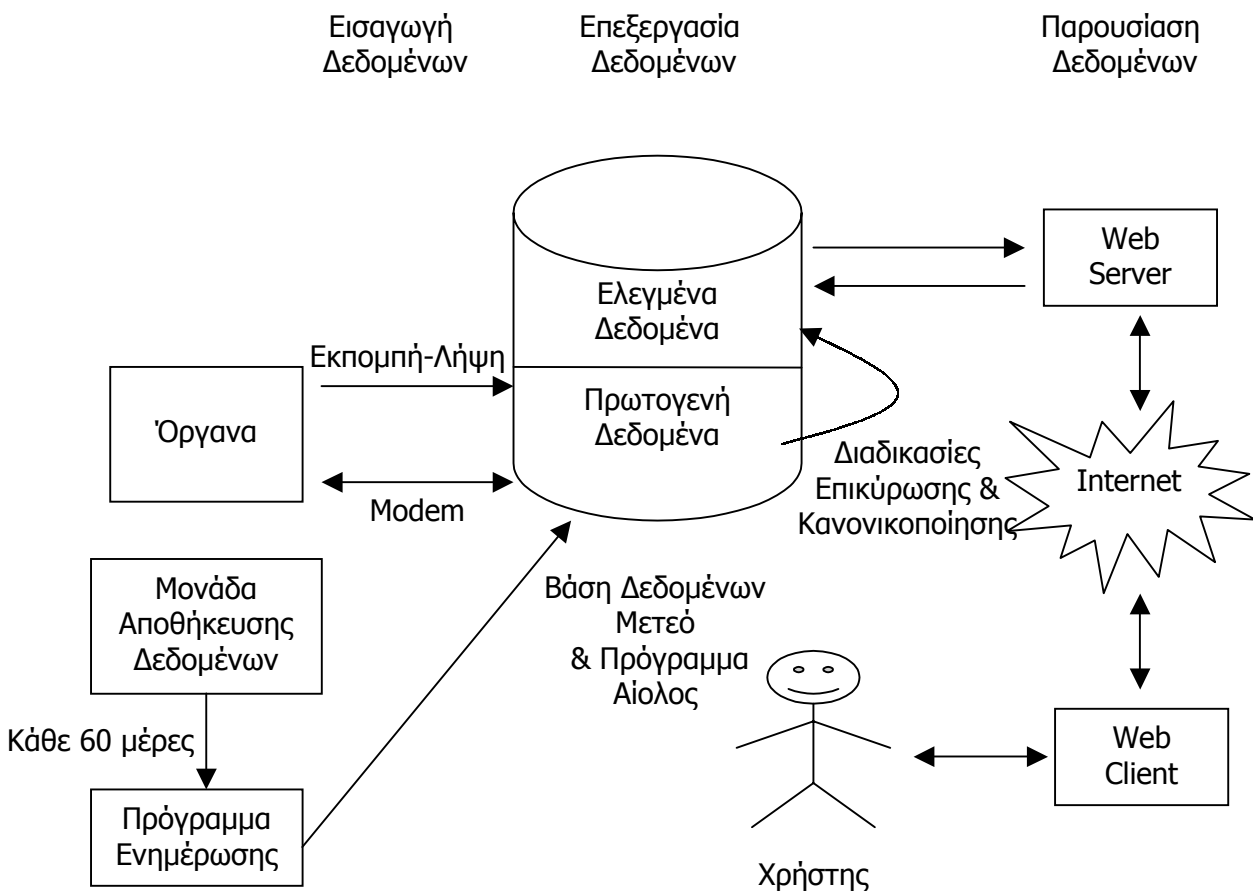
2.1.3.3 Στάδιο παρουσίασης δεδομένων

Στο τελευταίο στάδιο της παρουσίασης των δεδομένων, ο Αίολος καλείται να δώσει μία φιλική στο χρήστη διεπαφή, αποτελούμενη από απλές ρουτίνες χειρισμού των δεδομένων που έχει επεξεργαστεί για να μπορούν να τις καλούν προγράμματα που αναλαμβάνουν τη γραφική απεικόνιση των αποτελεσμάτων (όπως είναι ένας web browser για παράδειγμα), αρκετά φιλική στον χρήστη. Ο Αίολος θα πρέπει να έχει τις παρακάτω προδιαγραφές:

- Παρουσίαση της τρέχουσας μέτρησης των οργάνων
- Πρόσβαση στο ιστορικό αρχείο του σταθμού
- Στατιστικά στοιχεία του κάθε αισθητήρα για τις τελευταίες 24 ώρες
- Παρουσίαση διαγραμμάτων του κάθε αισθητήρα για τις τελευταίες 24 ώρες
- Παρουσίαση διαφόρων άλλων πληροφοριών για τον σταθμό, επικοινωνία με την ομάδα και απαντήσεις συχνών ερωτημάτων.

2.1.3.4 Μοντέλο σταδίων του Αίολου

Στο παρακάτω σχήμα δίνεται το μοντέλο της αλληλεπίδρασης του Αίολου με τους άλλους παράγοντες. Στο σχήμα αυτό μπορούμε να δούμε τα τρία στάδια που προαναφέρθηκαν και τον τρόπο με τον οποίο γίνονται οι επαφές με τον εξωτερικό κόσμο. Παρατηρούμε στο πρώτο στάδιο τις τρεις διαφορετικές εισόδους δεδομένων, οι οποίες λαμβάνονται από το σύστημα, που φροντίζει την επεξεργασία τους, την καταχώρηση και τη παρουσίαση στον χρήστη.



2.1.4 Ορισμός λειτουργικών απαιτήσεων

Ο Αίολος θέλουμε να κάνει μερικές συγκεκριμένες λειτουργίες οι οποίες αναφέρθηκαν και προηγουμένως. Σε αυτό το σημείο συνοψίζονται τρεις κύριες λειτουργίες:

- Θα έχει τρεις διαφορετικές εισόδους δεδομένων

- Θα επεξεργάζεται τα δεδομένα και θα παράγει αποτελέσματα, τα οποία θα καταχωρούνται στη βάση μαζί με τα πρωτογενή δεδομένα
- Θα ικανοποιεί αιτήσεις για παρουσίαση και επεξεργασία δεδομένων

Ένα λειτουργικό σενάριο είναι το ακόλουθο:

Ο Αίολος κάθε 10 λεπτά θα δέχεται δεδομένα από δύο διαφορετικές πηγές, με δύο διαφορετικούς τρόπους (με ασύρματη ζεύξη και με τηλεφωνική σύνδεση modem). Κάθε δύο μήνες περίπου, ο Αίολος θα πρέπει να είναι σε θέση να δεχτεί τα δεδομένα που έχουν αποθηκευτεί στη μονάδα αποθήκευσης που είναι στο σταθμό, με σκοπό να τα εισάγει ή καλύτερα να κάνει έλεγχο με τα δεδομένα που λήφθηκαν ασύρματα και να διορθώνει τυχόν λάθος εγγραφές. Ο έλεγχος πάνω και κάτω ορίων που γίνεται σε όλες τις τιμές, είναι αυστηρός, με αποτέλεσμα αν βρεθεί μία λάθος μέτρηση σε ένα όργανο μηδενίζεται σχετική εγγραφή. Συνεπώς τα δεδομένα του συλλέκτη, δεν έχουν λάθη λόγω μετάδοσης και θεωρούνται σωστά.

Στη συνέχεια θα γίνεται απευθείας εισαγωγή των δεδομένων στη βάση. Η επεξεργασία τους θα γίνεται κάθε 10 λεπτά της ώρας, κάθε 1 ώρα, κάθε μέρα, κάθε μήνα και κάθε χρόνο, σε χρόνους που δεν θα εξαρτιούνται από τον χρόνο με τον οποίο έρχονται τα δεδομένα. Θα γίνεται δηλαδή μία κανονικοποίηση των δεδομένων για καλύτερη παρουσίαση. Προφανώς οι κανονικοποιημένες τιμές θα πρέπει να καταχωρούνται σε διαφορετικό πίνακα, από αυτόν των πρωτογενών δεδομένων.

Η παρουσίαση των δεδομένων στον χρήστη θα γίνεται μέσω του World Wide Web και με ένα πρόγραμμα πλοήγησης (browser). Σε αυτό θα πρέπει να παρέχονται όλες οι ευκολίες και δυνατότητες στον χρήστη, για την πρόσβασή του στα δεδομένα. Θα γίνεται και αποστολή δεδομένων στον χρήστη, από το αρχείο των μετρήσεων που θα είναι καταγραμμένο στη βάση.

2.1.5 Ορισμός μη λειτουργικών απαιτήσεων

Σε αυτό το κομμάτι θα ορίσουμε όλες τις μη λειτουργικές απαιτήσεις, δηλαδή όλους του περιορισμούς στην υλοποίηση του λογισμικού και της σχεδίασης, καθώς και λεπτομέρειες παρουσίασης των δεδομένων, χρόνου απόκρισης και οτιδήποτε άλλο μπορεί να χρειαστεί να ληφθεί υπόψη.

- Η βάση δεδομένων θα πρέπει να τρέχει σε ισχυρό και γρήγορο υπολογιστή, γιατί θα είναι μεγάλος ο φόρτος από τις παράλληλες εισαγωγές, επεξεργασίες των δεδομένων και την εκπλήρωση αιτημάτων για παρουσίαση.
- Η παρουσίαση των δεδομένων θα πρέπει να έχει μία αρκετά καλή διεπαφή, που να είναι φιλική για το χρήστη.
- Ο χρήστης θα μπορεί να ζητήσει και να λάβει αυτόματα δεδομένα μέσω του δικτύου, για το σκοπό αυτό θα προσδιορίζει τη χρονική διακριτότητα των δεδομένων που θέλει να λάβει, δηλαδή δεκάλεπτη, ωριαία ή ημερήσια, καθώς και τη χρονική περίοδο που επιθυμεί. Τα δεδομένα θα στέλνονται σε αρχεία ASCII.

2.1.6 Εξέλιξη συστήματος

Ο Αίολος θα πρέπει να μεριμνήσει για όλες τις πιθανές εξελίξεις που μπορεί να έχει το σύστημα. Παρακάτω αναφέρονται μερικές ενδεικτικές εξελίξεις:

- Θα πρέπει να μπορεί εύκολα ο Αίολος να προσαρμοστεί σε πιθανή προσθήκη νέου αισθητήρα μέτρησης
- Καλό κρίνεται να μπορεί να έχει εναλλακτικούς τρόπους εισαγωγής δεδομένων, που ενδεχομένως να απαιτηθούν στο μέλλον

Σκόπιμο θεωρείται να είναι ευέλικτη και η έξοδος των δεδομένων από τη βάση με μία τέτοια μορφοποίηση (π.χ. αρχείων ASCII), που να είναι γενική και να μπορεί να χρησιμοποιηθεί σε κάθε πλατφόρμα

2.2 Τρέχουσα τεχνολογική κατάσταση

Για την υλοποίηση απαιτήθηκε η αγορά και εγκατάσταση μίας βάσης δεδομένων. Επιλέχθηκε η Oracle σαν την πιο σταθερή, περισσότερο διαδεδομένη και υποστηριζόμενη από άλλες εφαρμογές. Ο όγκος των δεδομένων που καλείται να διαχειριστή είναι μεγάλος, της τάξης των εκατομμυρίων εγγραφών και η επιλογή του καλύτερου RDBMS ήταν αναγκαία. Η έκδοση της Oracle που χρησιμοποιήθηκε, είναι η 8.1.5 ή Oracle8i για Windows NT. Έχει βγει η 8.1.7 που όταν κυκλοφορήσει στην Ελλάδα, θα αντικαταστήσει την υπάρχουσα και θα μπει σε Windows 2000 server.

Η Oracle8i, η βάση δεδομένων για υπολογισμούς του διαδικτύου, αλλάζει τον τρόπο με τον οποίο η πληροφορία διαχειρίζεται και προσπελάζεται, για να συναντήσει τις απαιτήσεις της εποχής του διαδικτύου, καθώς προσφέρει σημαντικά νέα στοιχεία για τις παραδοσιακές εφαρμογές online transaction processing (OLTP) και data warehousing. Προμηθεύει προηγμένα εργαλεία για να διαχειρίζεται κανείς όλους τους τύπους των δεδομένων στα web sites, αλλά επίσης διαθέτει την απόδοση, την ευελιξία και την διαθεσιμότητα που απαιτείται για πολύ μεγάλες βάσεις δεδομένων (very large database VLDB) και κρίσιμες εφαρμογές (mission-critical applications).

Η Oracle8i είναι πολύ περισσότερη από μία απλή σχεσιακή αποθήκευση δεδομένων. Εισάγει το iFS, το Internet File System που επιτρέπει στους χρήστες να μετακινήσουν εύκολα όλα τους τα δεδομένα μέσα σε μία Oracle8i βάση δεδομένων, όπου μπορούν να αποθηκευτούν και να διαχειριστούν περισσότερο αποδοτικά σε μία ενοποιημένη μορφή. Μία νέα επιλογή, Oracle8i interMedia, επιτρέπει στις επιχειρήσεις να διαχειριστούν και να έχουν πρόσβαση σε multi-media δεδομένα, συμπεριλαμβανομένου εικόνας, αρχείου, ήχου, βίντεο και χωρικά δεδομένα. Το interMedia Clipboard και ο Web Agent, δουλεύουν μαζί για να καταστήσουν web ικανό το interMedia. Το WebDB είναι ένα νέο HTML βασισμένο εργαλείο ανάπτυξης για να δημιουργούνται ιστοσελίδες HTML με περιεχόμενο βασισμένο στα δεδομένα που έχουν αποθηκευτεί σε βάσεις δεδομένων Oracle.

Η Oracle8i εισάγει νέα υποστήριξη για την Java, συμπεριλαμβάνοντας ένα εύρωστο, ενοποιημένο και ευέλικτο Java Virtual Machine μέσα στον διακομιστή. Αυτό επεκτείνει την υποστήριξη της Oracle για την Java σε όλους τους δεσμούς των εφαρμογών, επιτρέποντας στα προγράμματα σε Java να αναπτύσσονται όπου εκτελούνται καλύτερα (μέσα στον πελάτη, στον διακομιστή ή στον ενδιάμεσο), χωρίς να μεταγλωττίζεται ή τροποποιείται ο κώδικας της Java.

Όχι μόνο η Oracle8i εισάγει σημαντική καινοτομία στην τεχνολογία, όπως είναι η Java VM και το iFS, αλλά επίσης έχει σημαντικά νέα χαρακτηριστικά και λειτουργικότητα για τις παραδοσιακές εφαρμογές OLTP και data warehousing. Για τα data warehouses, η Oracle8i τώρα παρέχει φιλοσοφημένα χαρακτηριστικά συνολικής διαχείρισης για να αποθηκεύονται σύνολα που ερωτούνται συχνά, μειώνοντας σημαντικά την επεξεργασία των ερωτήσεων.

Για εφαρμογές OLTP, η Oracle8i εισάγει το χαρακτηριστικό σταθερότητας, το σχέδιο του βελτιστοποιητή, που επιτρέπει στον χρήστη να επιβεβαιώσει ότι ο βελτιστοποιητής επιλέγει το ίδιο ρυθμισμένο σχέδιο κάθε φορά που εκτελείται η ίδια ερώτηση. Το Advanced Queueing έχει επαυξηθεί για να υποστηρίζει πρότυπο δημοσίευσης ή εγγραφής. Οι εφαρμογές OLTP επωφελούνται από πάρα πολλά νέα και επαυξημένα χαρακτηριστικά που βελτιώνουν την ανακτησιμότητα και διαθεσιμότητα της βάσης δεδομένων κατά την διάρκεια λειτουργιών ρουτίνας, όπως είναι η επανάληψη της δημιουργίας ευρετηρίου, σε καταστροφικές καταστάσεις όπου η εφεδρική βάση μπορεί να ενεργοποιηθεί.

Η Oracle8i διαθέτει την διαχείριση πόρων βάσης δεδομένων, όπου το DBA τώρα έχει την ικανότητα να ελέγχει τους καταμερισμένους πόρους της επεξεργασίας σε έναν χρήστη ή σε μία ομάδα χρηστών. Δύο νέοι μέθοδοι κατάτμησης, το hash και το composite, συμπληρώνουν την καθιερωμένη εμβέλεια της μεθόδου κατάτμησης, για να παρέχουν ένα πλούσιο σύνολο μεθόδων

κατάτμησης από το οποίο το DBA μπορεί να επιλέξει την καλύτερη μέθοδο που ταιριάζει στο προφίλ και στο φόρτο μίας εφαρμογής.

Ο Oracle Parallel Server συνθέτει μία νέα αρχιτεκτονική στην Oracle8i. Το cache fusion είναι μία νέα χωρίς δίσκο αρχιτεκτονική ring που βελτιώνει σημαντικά την επικοινωνία με το εσωτερικό στιγμιότυπο. Νέα χαρακτηριστικά δικτύου βελτιώνουν την ευκολία για τη χρήση του OPS και η διαχείριση του συστήματος έχει γίνει ευκολότερη μέσω της επαύξησης του Oracle Server Management, του νέου Oracle Universal Installer και του Oracle Database Configuration Assistant.

Η Oracle8i επεκτείνει την λειτουργικότητα της προηγμένης αντιγραφής, εστιάζοντας σε εφαρμογές μαζικής παραγωγής. Τα δεδομένα μπορούν να αντιγραφούν σε διακομιστές που είναι κοντύτερα σε χρήστες και έχουν μόνο τα δεδομένα που αυτοί οι χρήστες χρειάζονται, παρέχοντας καλύτερη απόδοση. Η ασφάλεια έχει βελτιωθεί. Ο συντονιστής αντιγραφής έχει ξαναγραφτεί σε Java και δεν περιορίζεται πια να τρέχει σε μηχανήματα που βασίζονται στα Windows.

Καθώς η Oracle έχει επεκτείνει την υποστήριξη της στα δεδομένα πολυμέσων μέσω του interMedia και άλλων σύνθετων τύπων δεδομένων μέσω ξεχωριστών πακεταρισμένων επιλογών Visual Information Retrieval, Time Series και Spatial, παρέχει επίσης μία νέα επεκτάσιμη αρχιτεκτονική για ανεξάρτητους προμηθευτές λογισμικού (independent software vendors) και άλλους δημιουργούς να εκμεταλλευτούν όταν αναπτύσσουν υποστήριξη για νέους τύπους δεδομένων ή εξειδικευμένες επεξεργασίες. Υπάρχει μία σημαντική επέκταση στην τεχνολογία σχέσεως της Oracle και η Oracle8i το κάνει διαθέσιμο στην standard και στην Enterprise έκδοση.

Βελτιώσεις της γλώσσας για PL/SQL, Pro*C/C++, Pro*COBOL και το Oracle Call Interface (OCI) συμπεριλαμβάνει μία νέα σημαντική λειτουργικότητα. Επίσης το National Language Support υφίσταται δυνατή αλλαγή και υπάρχουν πληροφορίες για το πώς θα την χρησιμοποιήσεις στην τεκμηρίωση.

Το τέλειο ίχνος ελέγχου πρόσβασης και το γενικό πλαίσιο χαρακτηριστικών των εφαρμογών, χτίζουν στη βάση δεδομένων μία ασφάλεια εφαρμογής σε επίπεδο γραμμής, παρά να το αφήνουν στο επίπεδο εφαρμογής. Μεγαλύτερη ασφάλεια παρέχεται σε όλη την επιχείρηση με την επέκταση ή την επαύξηση μεθόδων επικύρωσης και εξουσιοδότησης, ενοποιημένη διαχείριση χρηστών και υποστήριξη στάνταρ.

Τέλος η Oracle8i συμπεριλαμβάνει τον Oracle Enterprise Manager, που είναι το πλήρες πλαίσιο εργασίας για διαχείριση της βάσης δεδομένων Oracle και του περιβάλλοντος των εφαρμογών. Ο Oracle Enterprise Manager παρουσιάζει μία εύκολη στη χρήση κεντροποιημένη κονσόλα, ένα πλούσιο σύνολο από εργαλεία διαχείρισης και την επεκτασιμότητα να εντοπίζει και να επιλύει οποιοδήποτε πρόβλημα που μπορεί να εμφανιστεί. Επίσης συμπεριλαμβάνει αρκετές διαχειριστικές εφαρμογές για να πραγματοποιεί καθημερινές εργασίες για τις βάσεις δεδομένων και τις εφαρμογές, όπως είναι η πραγματοποίηση ρουτινών backup.

3 ΚΕΦΑΛΑΙΟ 3: ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

3.1 Περιγραφή αρχιτεκτονικής της βάσης δεδομένων

3.1.1 Εισαγωγή

Σε αυτό το κεφάλαιο αναλύεται λεπτομερειακά ο σχεδιασμός της βάσης δεδομένων. Θα παρουσιαστεί το λογικό και το φυσικό επίπεδο.

3.1.2 Λογικό Σχήμα

Το λογικό επίπεδο της βάσης αποτελείται από τις παρακάτω οντότητες:

- Σταθμός (Station)
- Βίντεο (Video)
- Φωτογραφία (Photo)
- Τύπος Σταθμού (Station Type)
- Λόγος (Reason)
- Μη Διαθεσιμότητα Αισθητήρα (Sensor NotAvail)
- Αισθητήρας (Sensor)
- Μεταβλητή (Variable)
- Χρονικό Βήμα (Time Step)
- Είδος Χρονοσειράς (Series Type)
- Χρονοσειρά (Time Series)
- Πρωτογενή Δεδομένα (Data Raw)
- Επεξεργασμένα Δεδομένα (Data Proc)
- Συνθετικά Δεδομένα (Data Synth)

Παρακάτω αναλύονται οι οντότητες που προαναφέρθηκαν.

3.1.2.1 Σταθμός (Station)

Η οντότητα σταθμός περιγράφει τον σταθμό στον οποίο βρίσκεται ο αισθητήρας. Τα χαρακτηριστικά αυτής της οντότητας δίνονται παρακάτω:

- Κωδικός Σταθμού (Station ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Θέση (Position)
- Κωδικός Τύπου (Type ID)
- Σχόλια (Comments)

Ο Κωδικός είναι ένας μοναδικός αριθμός για το πρωτεύον κλειδί της οντότητας. Το Όνομα είναι η ονομασία του σταθμού και το Όνομα στα ελληνικά είναι το αντίστοιχο όνομα στα ελληνικά. Η θέση του σταθμού αποθηκεύεται στο αντίστοιχο πεδίο. Ο Τύπος του σταθμού, δίνει την λειτουργία που έχει αυτός ο σταθμός. Το πεδίο Σχόλια έχει επιπλέον πληροφορίες για τον σταθμό.

3.1.2.2 Βίντεο (Video)

Η οντότητα Βίντεο, δίνει τα βίντεο που μπορεί να έχει ο κάθε σταθμός. Τα χαρακτηριστικά αυτής της οντότητας είναι:

- Κωδικός Βίντεο (Video ID)
- Κωδικός Σταθμού (Station ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Μονοπάτι (Pathname)
- Σχόλια (Comments)

Ο Κωδικός Βίντεο είναι ένας μοναδικός αριθμός που χαρακτηρίζει το συγκεκριμένο βίντεο. Ο Κωδικός Σταθμού είναι το αναγνωριστικό του σταθμού, στον οποίο ανήκει το συγκεκριμένο βίντεο. Το Όνομα κρατάει την αγγλική περιγραφή του βίντεο και το Ελληνικό Όνομα την ελληνική. Το Μονοπάτι, δίνει την διαδρομή του καταλόγου του λειτουργικού συστήματος και τα Σχόλια δίνουν περισσότερες επεξηγηματικές πληροφορίες για το συγκεκριμένο βίντεο.

3.1.2.3 Φωτογραφία (Photo)

Η οντότητα Φωτογραφία, όπως και η οντότητα Βίντεο, κρατάει τις φωτογραφίες που μπορεί να έχει ένας σταθμός. Τα χαρακτηριστικά πεδία της σχέσης είναι:

- Κωδικός Φωτογραφίας (Photo ID)
- Κωδικός Σταθμού (Station ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Μονοπάτι (Pathname)
- Σχόλια (Comments)

Το αναγνωριστικό της φωτογραφίας, αποθηκεύεται στον Κωδικό της Φωτογραφίας. Η σχέση με τον σταθμό βγαίνει από τον Κωδικό του Σταθμού. Αγγλική και ελληνική ονομασία φυλάγεται στο Όνομα και στο Ελληνικό όνομα αντίστοιχα. Το Μονοπάτι έχει την διαδρομή της τοποθεσίας της φωτογραφίας και τα Σχόλια παρέχουν περισσότερες πληροφορίες.

3.1.2.4 Τύπος Σταθμού (Station Type)

Η οντότητα Τύπος Σταθμού, περιγράφει τον τύπο στον οποίο ανήκει ένας σταθμός. Τα χαρακτηριστικά της είναι:

- Κωδικός Τύπου (Type ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Σχόλια (Comments)

Ο Κωδικός είναι το αναγνωριστικό του τύπου, τα δύο Ονόματα είναι η περιγραφή του σε δύο γλώσσες και τα σχόλια για εκτενέστερη πληροφορία.

3.1.2.5 Αιτία (Reason)

Η οντότητα Αιτία, περιγράφει όλες τις δυνατές περιπτώσεις που μπορεί να έχουμε μη διαθεσιμότητα μετρήσεων από κάποιον αισθητήρα. Τα χαρακτηριστικά της είναι:

- Κωδικός Αιτίας (Reason ID)
- Περιγραφή (Description)
- Ελληνική Περιγραφή (GR Description)
- Σχόλια (Comments)

Ο Κωδικός είναι το αναγνωριστικό κλειδί της κάθε αιτίας. Δίνονται δύο περιγραφές στα αγγλικά και στα ελληνικά από την Περιγραφή και την Ελληνική Περιγραφή αντίστοιχα. Τα Σχόλια κρατάν επιπλέον χρήσιμες πληροφορίες.

3.1.2.6 Μη Διαθεσιμότητα Αισθητήρα (Sensor NotAvail)

Η σχέση της μη διαθεσιμότητας ενός αισθητήρα και της μεταβλητής για τον οποία δεν υπάρχουν μετρήσεις στο συγκεκριμένο χρονικό διάστημα, την δίνει η Μη Διαθεσιμότητα Αισθητήρα. Τα χαρακτηριστικά αυτής της σχέσης είναι:

- Κωδικός Αισθητήρα (Sensor ID)
- Κωδικός Μεταβλητής (Variable ID)
- Ημερομηνία Έναρξης (Start Date)
- Ημερομηνία Λήξης (End Date)
- Κωδικός Αιτίας (Reason ID)
- Σχόλια (Comments)

Η σχέση Μη διαθεσιμότητα Αισθητήρα, συσχετίζει των Κωδικό του Αισθητήρα και τον Κωδικό της Μεταβλητής, μαζί με την Ημερομηνία Έναρξης της περιόδου μη διαθεσιμότητας. Επιπλέον πεδία που δίνονται είναι η Ημερομηνία Λήξης της μη διαθεσιμότητας, ο Κωδικός Αιτίας που εξηγεί τον λόγο για τον οποίο δεν υπάρχουν μετρήσεις και τα επεξηγηματικά Σχόλια.

3.1.2.7 Αισθητήρας (Sensor)

Ο αισθητήρας αποτελείται από τα παρακάτω χαρακτηριστικά:

- Κωδικός Αισθητήρα (Sensor ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Κωδικός Σταθμού (Station ID)
- Μάρκα (Firm)
- Μοντέλο (Model)
- Ημερομηνία Έναρξης (Start Date)
- Ημερομηνία Λήξης (End Date)
- Κατάσταση (Status)
- Ακρίβεια (Accuracy)
- Σχόλια (Comments)
- Ελληνικά Σχόλια (GR Comments)

Ο κωδικός του αισθητήρα είναι το πρωτεύον κλειδί της οντότητας και είναι ένας αριθμός μοναδικός για τον κάθε αισθητήρα. Το Όνομα αναφέρεται στον τίτλο που περιγράφει τον συγκεκριμένο αισθητήρα και το ελληνικό όνομα είναι ο τίτλος στα ελληνικά. Ο κωδικός Σταθμού είναι το ξένο κλειδί του πίνακα Σταθμού που περιγράφτηκε προηγουμένως. Η Μάρκα είναι η μάρκα του αισθητήρα και το μοντέλο του το διατηρεί το χαρακτηριστικό Μοντέλο. Η Ημερομηνία Έναρξης και Λήξης, αναφέρεται στην ημερομηνία που ο αισθητήρας ξεκίνησε και σταμάτησε να καταγράφει αντίστοιχα. Η Κατάσταση του αισθητήρα είναι ένα πεδίο στο οποίο θα δηλώνεται με έναν χαρακτήρα λατινικό αν είναι ενεργός, ανενεργός, χαλασμένος κλπ. Τέλος στο χαρακτηριστικό

Σχόλια κρατιούνται όλες οι χρήσιμες πληροφορίες για τον κάθε αισθητήρα και στα Ελληνικά Σχόλια το ίδιο, αλλά στα ελληνικά.

3.1.2.8 Μεταβλητή (Variable)

Στην οντότητα Μεταβλητή αποθηκεύονται όλες οι μετεωρολογικές μεταβλητές που χρησιμοποιεί το σύστημα. Τα χαρακτηριστικά αυτής της οντότητας δίνονται παρακάτω:

- Κωδικός Μεταβλητής (Variable ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Μονάδα (Unit)
- Σχόλια (Comments)

Ο Κωδικός της Μεταβλητής είναι ένας μοναδικός αριθμός που χαρακτηρίζει την κάθε μετεωρολογική μεταβλητή. Το Όνομα και το Ελληνικό Όνομα είναι το όνομα που έχει η μεταβλητή στα αγγλικά και ελληνικά αντίστοιχα, όπως για παράδειγμα η θερμοκρασία. Η Μονάδα είναι το σύμβολο της μονάδας μέτρησης στην οποία μετράμε την αντίστοιχη μετεωρολογική μεταβλητή. Στα Σχόλια καταχωρούνται όλες οι επιπρόσθετες πληροφορίες για την κάθε μεταβλητή.

3.1.2.9 Χρονικό Βήμα (Time Step)

Το Χρονικό Βήμα αναφέρεται στο βήμα που έχει η κάθε χρονοσειρά. Τα χαρακτηριστικά αυτής της οντότητας είναι:

- Κωδικός Βήματος (Step ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Σχόλια (Comments)

Ο Κωδικός του Χρονικού Βήματος είναι ένας αναγνωριστικός αριθμός, μοναδικός για το κάθε βήμα. Το Όνομα και Ελληνικό Όνομα είναι η ονομασία του βήματος που ακολουθεί η χρονοσειρά στα αγγλικά και ελληνικά αντίστοιχα. Τέλος τα Σχόλια κρατούν επιπλέον πληροφορίες που αφορούν το χρονικό βήμα.

3.1.2.10 Είδος Χρονοσειράς (Series Types)

Το Τύπος της Χρονοσειράς περιγράφει το είδος που έχει η κάθε χρονοσειρά που παράγεται από τα πρωτογενή ή τα κανονικοποιημένα δεδομένα. Τα χαρακτηριστικά της είναι:

- Κωδικός Τύπου (Type ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Σχόλια (Comments)

Ο Κωδικός είναι το κλειδί και αναφέρεται σε ένα μοναδικό αριθμό για το κάθε τύπο χρονοσειράς που μπορεί να έχουμε. Το Όνομα δίνει ένα πιο κατανοητό από τον άνθρωπο όνομα για τον τύπο της χρονοσειράς και το Ελληνικό Όνομα δίνει το ίδιο, αλλά γραμμένο στα ελληνικά. Στα Σχόλια μπορούν να αποθηκευτούν επιπλέον πληροφορίες για μία συγκεκριμένη χρονοσειρά.

3.1.2.11 Χρονοσειρά (Time Series)

Η Χρονοσειρά είναι μία οντότητα που συσχετίζει όλες τις τιμές των μετεωρολογικών ακολουθιών που προκύπτουν από την επεξεργασία των δεδομένων σε μία ακολουθία δεδομένων. Παρακάτω δίνονται τα χαρακτηριστικά της οντότητας:

- Κωδικός Σειράς (Series ID)
- Όνομα (Name)
- Ελληνικό Όνομα (GR Name)
- Κωδικός Μεταβλητής (Variable ID)
- Κωδικός Βήματος (Step ID)
- Κωδικός Αισθητήρα (Sensor ID)
- Κωδικός Τύπου (Type ID)
- Σχόλια Σειράς (Comments)

Ο Κωδικός της Σειράς είναι ένας μοναδικός αναγνωριστικός αριθμός της χρονοσειράς που έχει και το ρόλο του πρωτεύοντος κλειδιού. Διαθέτει δύο ονόματα γραμμένα στην αγγλική και την ελληνική γλώσσα. Οι Κωδικοί Αισθητήρα, Βήματος, Προέλευσης, Μεταβλητής και Τύπου είναι τα ξένα κλειδιά των οντοτήτων που προαναφέρθηκαν. Τα Σχόλια κρατούν χρήσιμες επιπλέον πληροφορίες για τη χρονοσειρά.

3.1.2.12 Πρωτογενή Δεδομένα (Data Raw)

Η οντότητα Δεδομένα έχει όλα τα δεδομένα που καταγράφουν τα όργανα κάθε 10 λεπτά. Τα χαρακτηριστικά αυτής της οντότητας φαίνονται παρακάτω:

- Ημερομηνία Δεδομένων (Data Date)
- Κωδικός Χρονοσειράς (Series ID)
- Κατάσταση (Status)
- Τιμή (Value)

Η Ημερομηνία των Δεδομένων είναι ένας αριθμός που περιγράφει την ώρα και την ημερομηνία που ελήφθησαν τα δεδομένα και καταχωρήθηκαν στον Αίολο. Ο Κωδικός της Χρονοσειράς είναι αναγνωριστικό της χρονοσειράς στην οποία συμμετέχουν τα δεδομένα και μαζί με την ημερομηνία αποτελούν το πρωτεύον κλειδί. Η Κατάσταση των Δεδομένων είναι ένας χαρακτήρας που περιγράφει την κατάσταση στην οποία βρίσκονται τα δεδομένα. Αυτή μπορεί να περιγράφει για παράδειγμα την εγκυρότητα των δεδομένων ή αν είναι ελεγμένα κλπ. Η Τιμή των Δεδομένων είναι η δεκαδική τιμή του οργάνου μέτρησης.

3.1.2.13 Επεξεργασμένα Δεδομένα (Data Proc)

Τα δεδομένα που προέρχονται από κάποια επεξεργασία των πρωτογενών, όπως για παράδειγμα είναι ο έλεγχος ακραίων τιμών και χρονικής συνέπειας, αποθηκεύονται σε ξεχωριστή οντότητα. Η οντότητα Επεξεργασμένα Δεδομένα έχει:

- Ημερομηνία Δεδομένων (Data Date)
- Κωδικός Χρονοσειράς (Series ID)
- Κατάσταση (Status)
- Τιμή (Value)

Η εξήγηση των πεδίων αυτών έχει δοθεί στην προηγούμενη παράγραφο.

3.1.2.14 Συνθετικά Δεδομένα (Data Synth)

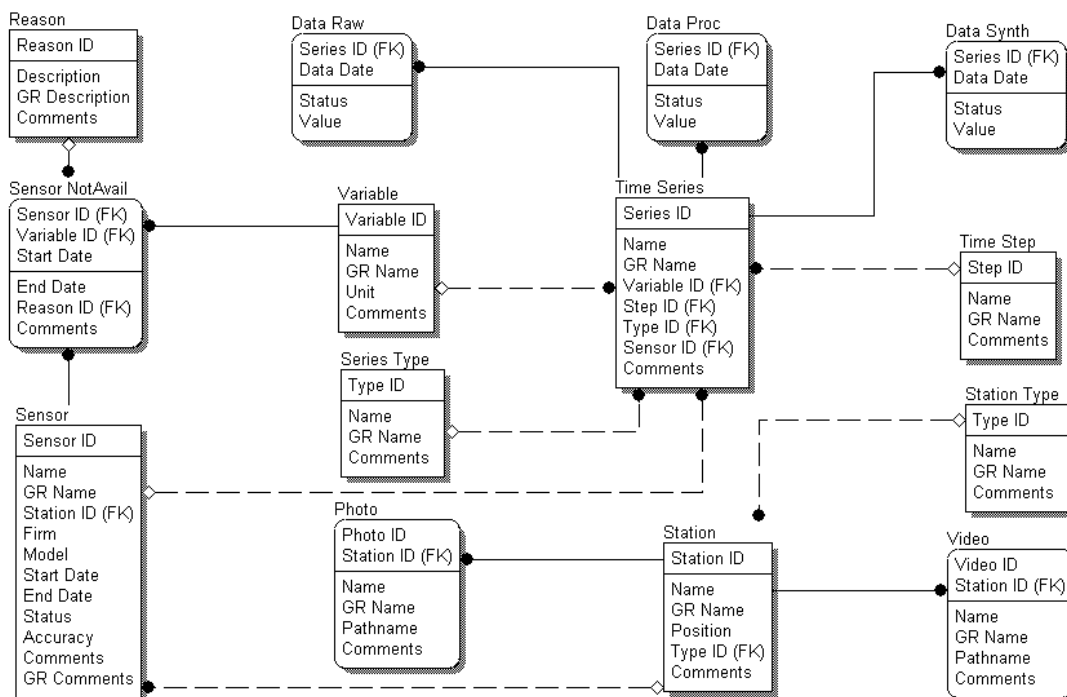
Η οντότητα Συνθετικά Δεδομένα φυλάει τα δεδομένα που προέρχονται από την σύνθεση δύο ή περισσότερων χρονοσειρών, για να προκύψει μία συμπληρωμένη ή μία εντελώς καινούργια χρονοσειρά. Τα πεδία της είναι:

- Ημερομηνία Δεδομένων (Data Date)
- Κωδικός Χρονοσειράς (Series ID)
- Κατάσταση (Status)
- Τιμή (Value)

Είναι ακριβώς ίδια με τις δύο προηγούμενες οντότητες και η επεξήγηση των χαρακτηριστικών έχει γίνει στην πρώτη από αυτές.

3.1.2.15 Σχήμα Οντοτήτων – Συσχετίσεων

Το σχήμα Οντοτήτων – Συσχετίσεων που προκύπτει για το παραπάνω μοντέλο βάσης δεδομένων που αναφέρθηκε είναι:

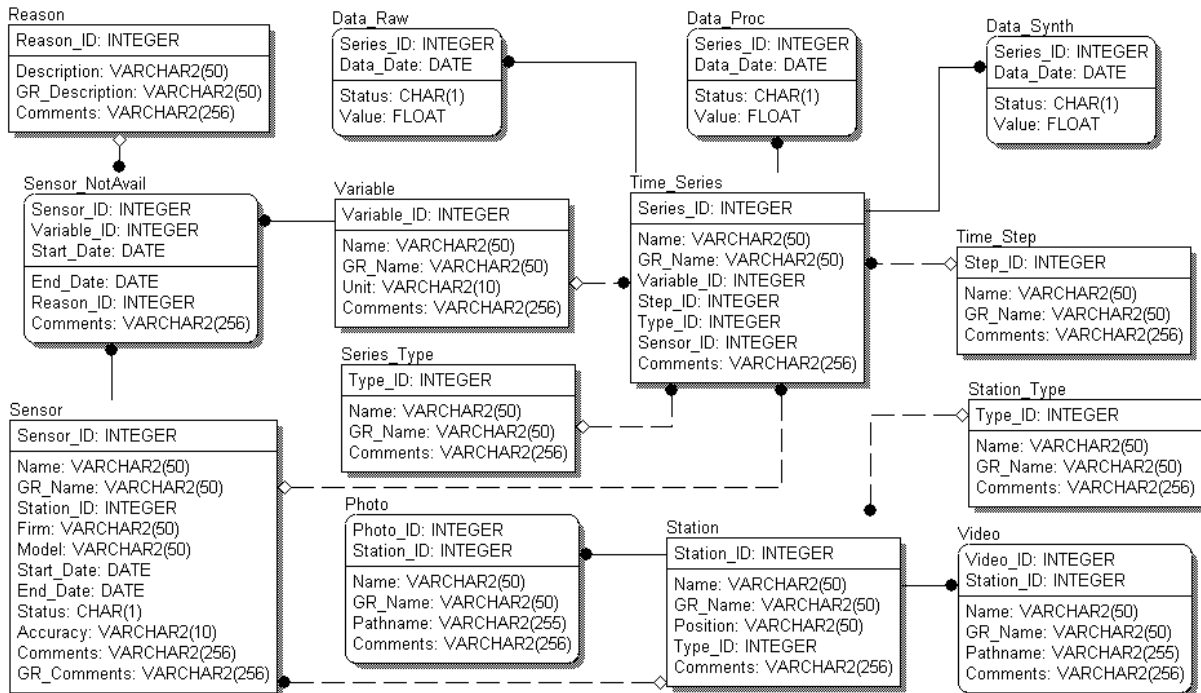


3.1.3 Φυσικό σχήμα

Στο φυσικό σχήμα της βάσης γίνεται μία ανάλυση των χαρακτηριστικών της κάθε οντότητας και παρουσιάζονται τα ονόματα των πινάκων και των δηλώσεων που έχουν τα χαρακτηριστικά. Οι πίνακες που έχει η βάση είναι:

- Station (Station_ID, Name, GR_Name, Position, Type_ID, Comments)
- Video (Video_ID, Station_ID, Name, GR_Name, Pathname, Comments)
- Photo (Photo_ID, Station_ID, Name, GR_Name, Pathname, Comments)
- Reason (Reason_ID, Description, GR_Description, Comments)
- Sensor_NotAvail (Sensor_ID, Variable_ID, Start_Date, End_Date, Reason_ID, Comments)
- Sensor (Sensor_ID, Name, GR_Name, Station_ID, Firm, Model, Start_Date, End_Date, Status, Accuracy, Comments, GR_Comments)
- Variable (Variable_ID, Name, GR_Name, Unit, Comments)
- Series_Type (Type_ID, Name, GR_Name, Comments)
- Time_Step (Step_ID, Name, GR_Name, Comments)
- Time_Series (Series_ID, Name, GR_Name, Variable_ID, Step_ID, Type_ID, Sensor_ID, Comments)
- Data_Raw (Series_ID, Data_Date, Status, Value)
- Data_Proc (Series_ID, Data_Date, Status, Value)
- Data_Synth (Series_ID, Data_Date, Status, Value)

Παρακάτω δίνεται το διάγραμμα οντοτήτων - συσχετίσεων στο φυσικό επίπεδο της βάσης, όπου φαίνονται και οι δηλώσεις των τύπων των αντίστοιχων χαρακτηριστικών:



Από το παραπάνω διάγραμμα μπορούμε να δούμε τον τύπο δεδομένων για κάθε χαρακτηριστικών των οντοτήτων και των σχέσεων, αλλά θα τα παρουσιάσουμε και αναλυτικά με δίνοντας παραδείγματα για τις πιθανές τιμές που μπορούν να πάρουν.

3.1.3.1 Πίνακας Σταθμού (Station Table)

Οι τύποι δεδομένων των χαρακτηριστικών του πίνακα Σταθμού είναι:

- Station_ID: INTEGER
- Name: VARCHAR2(50)

- GR_Name: VARCHAR2(50)
- Position: VARCHAR2(50)
- Type_ID: INTEGER
- Comment: VARCHAR2(256)

Το πρώτο είναι ένας ακέριος που χαρακτηρίζει τον σταθμό. Το Name και το GR_Name είναι το όνομα που έχει δοθεί στον σταθμό εκφρασμένο σε αγγλικά και ελληνικά. Η τοποθεσία είναι οι συντεταγμένες του μετεωρολογικού σταθμού και το όνομα της τοποθεσίας. Ο τύπος εκφράζει τον τύπο του μετεωρολογικού σταθμού, αν δηλαδή είναι μετεωρολογικός ή κάτι άλλο. Στα σχόλια μπορεί να γράψει κανείς επιπλέον πληροφορίες και σημειώσεις για τον σταθμό.

3.1.3.2 Πίνακας Βίντεο (Video Table)

Ο πίνακας Βίντεο έχει τα παρακάτω πεδία:

- Video_ID: INTEGER
- Station_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Pathname: VARCHAR2(255)
- Comments: VARCHAR2(256)

Το Video_ID είναι ο ακέριος κωδικός του κάθε βίντεο. Το Station_ID είναι ο κωδικός του Σταθμού στον οποίο ανήκει το συγκεκριμένο βίντεο. Τα Name και GR_Name είναι η ονομασία του βίντεο στις δύο γλώσσες και το Pathname είναι η συμβολοσειρά με την πλήρη διαδρομή του βίντεο μέσα στο σύστημα αρχείων. Στο Comments μπορεί να γράψει κανείς μέχρι 256 χαρακτήρες επιπλέον σχόλια.

3.1.3.3 Πίνακας Φωτογραφίας (Photo Table)

Ο πίνακας Φωτογραφίας έχει τα παρακάτω πεδία:

- Photo_ID: INTEGER
- Station_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Pathname: VARCHAR2(255)
- Comments: VARCHAR2(256)

Το Photo_ID είναι ο ακέριος κωδικός της κάθε φωτογραφίας. Το Station_ID είναι ο κωδικός του Σταθμού στον οποίο ανήκει η συγκεκριμένη φωτογραφία. Τα Name και GR_Name είναι η ονομασία της φωτογραφίας στις δύο γλώσσες και το Pathname είναι η συμβολοσειρά με την πλήρη διαδρομή της φωτογραφίας. Στο Comments μπορεί να γράψει κανείς μέχρι 256 χαρακτήρες επιπλέον σχόλια.

3.1.3.4 Πίνακας Τύπου Σταθμού (Station_Type Table)

Ο πίνακας του Τύπου Σταθμού έχει τα παρακάτω πεδία:

- Type_ID: INTEGER
- Name: VARCHAR2(50)

- GR_Name: VARCHAR2(50)
- Comments: VARCHAR2(256)

Το Type_ID είναι ένας ακέραιος που διακρίνει τον κάθε τύπο σταθμού. Το Name είναι η ονομασία του τύπου στα αγγλικά και το GR_Name στα ελληνικά. Για σχόλια υπάρχει το Comments.

3.1.3.5 Πίνακας Αιτίας (Reason Table)

Ο πίνακας αυτός έχει τα παρακάτω πεδία μαζί με τον ορισμό τους:

- Reason_ID: INTEGER
- Description: VARCHAR2(50)
- GR_Description: VARCHAR2(50)
- Comments: VARCHAR2(256)

Το Reason_ID είναι το ακέραιο αναγνωριστικό του λόγου που ένας αισθητήρας είναι εκτός λειτουργίας. Τα υπόλοιπα πεδία είναι συμβολοσειρές που περιγράφουν την αιτία, δίνοντας το όνομα και σχόλια.

3.1.3.6 Πίνακας Μη Διαθεσιμότητας Αισθητήρα (Sensor_NotAvail Table)

Ο πίνακας αυτός έχει τα παρακάτω πεδία μαζί με τον ορισμό τους:

- Sensor_ID: INTEGER
- Variable_ID: INTEGER
- Start_Date: DATE
- End_Date: DATE
- Reason_ID: INTEGER
- Comments: VARCHAR2(256)

Συσχετίζονται οι ακέραιοι κωδικοί του αισθητήρα και της μετεωρολογικής μεταβλητής με την ημερομηνία έναρξης του χρονικού διαστήματος μη διαθεσιμότητας. Με το End_Date δηλώνεται η λήξη της έλλειψης μετρήσεων, για το οποίο έχει αιτιολογηθεί από το Reason_ID. Για επιπλέον πληροφορίες τα Comments, μπορούν να τα κρατήσουν.

3.1.3.7 Πίνακας Αισθητήρα (Sensor Table)

Για τον πίνακα Αισθητήρα έχουμε:

- Sensor_ID: INTEGER
- Name: VARCHAR2(30)
- GR_Name: VARCHAR2(30)
- Station_ID: INTEGER
- Firm: VARCHAR2(30)
- Model: VARCHAR2(30)
- Start_Date: Date
- End_Date: Date
- Status: CHAR(1)
- Accuracy: VARCHAR2(10)
- Comments: VARCHAR2(256)

- GR_Comments: VARCHAR2(256)

Το Sensor_ID που είναι ακέραιος θα παίρνει έναν αύξοντα αριθμό για κάθε καινούργιο όργανο. Τα υπόλοιπα είναι συμβολοσειρές που περιγράφουν τον αισθητήρα δίνοντας το όνομά του σε δύο γλώσσες, την μάρκα και το μοντέλο. Ο κάθε αισθητήρας ανήκει σε έναν σταθμό και φέρει το Station_ID του. Το Status παίρνει τιμή έναν λατινικό χαρακτήρα, για πιθανές καταστάσεις που μπορεί να έχει ένας αισθητήρας (A: Active, I: Inactive, D: Damaged). Στον αισθητήρα, μπορούν να μπου αγγλικά και ελληνικά σχόλια στα πεδία Comments και GR_Comments αντίστοιχα.

3.1.3.8 Πίνακας Μεταβλητής (Variable Table)

Για τον πίνακα Μεταβλητής, δίνονται οι τύποι των χαρακτηριστικών του παρακάτω:

- Variable_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Unit: VARCHAR2(10)
- Comments: VARCHAR2(256)

Το Variable_ID είναι ένα ακέραιο αναγνωριστικό της μεταβλητής και τα υπόλοιπα πεδία είναι συμβολοσειρές για πληροφορίες σχετικά με το είδος της μεταβλητής. Πιθανές τιμές για το Name είναι "Temperature", "Solar Radiation", "Rainfall", "Wind Direction" κλπ και για το GR_Name είναι αντίστοιχα «Θερμοκρασία», «Ηλιακή Ακτινοβολία», «Βροχόπτωση», «Διεύθυνση Ανέμου» κλπ. Για την μονάδα μέτρησης, χρησιμοποιείται το πεδίο Unit με 10 χαρακτήρες το μέγιστο και παίρνει τιμές '°C', 'min', 'm/s', 'W/m²' κλπ. Μέχρι 256 χαρακτήρες σχόλια για την κάθε μεταβλητή, επιτρέπονται στο πεδίο Comments.

3.1.3.9 Πίνακας Χρονικού Βήματος (Series_Type Table)

Για τον πίνακα Χρονικού Βήματος θα είναι:

- Step_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Comments: VARCHAR2(256)

Το πρώτο είναι ακέραιο αναγνωριστικό και τα υπόλοιπα συμβολοσειρές που περιγράφουν το χρονικό βήμα. Πιθανές τιμές για το Name είναι "Daily", "Monthly", "Yearly" κλπ. και για το GR_Name «Ημερήσια», «Μηνιαία», «Ετήσια» αντίστοιχα.

3.1.3.10 Πίνακας Τύπου Χρονοσειράς (Series_Type Table)

Ο πίνακας Τύπου Χρονοσειράς έχει τις παρακάτω δομές:

- Type_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Comments: VARCHAR2(256)

Το Type_ID είναι ένας μοναδικός ακέραιος και μαζί με τις συμβολοσειρές των δύο άλλων πεδίων περιγράφουν πλήρως τον τύπο της κάθε χρονοσειράς. Τιμές που μπορεί να πάρει το Name είναι "Ten-minutes", "Hourly", "Daily", "Monthly", "Yearly" και το GR_Name «Δεκάλεπτη», «Ωριαία», «Ημερήσια», «Μηνιαία» και «Ετήσια Χρονοσειρά» αντίστοιχα.

3.1.3.11 Πίνακας Χρονοσειράς (Time Series Table)

Για τον πίνακα της Χρονοσειράς έχουμε τα παρακάτω:

- Series_ID: INTEGER
- Name: VARCHAR2(50)
- GR_Name: VARCHAR2(50)
- Variable_ID: INTEGER
- Step_ID: INTEGER
- Type_ID: INTEGER
- Sensor_ID: INTEGER
- Comments: VARCHAR2(256)

Ο πίνακας Χρονοσειράς είναι μία σχέση μεταξύ των οντοτήτων που έχει η βάση και έτσι εκτός από το Series_ID που είναι κλειδί και αναγνωριστικό της κάθε χρονοσειράς, τα ονόματα και το Comments που έχει σχόλια για την αντίστοιχη χρονοσειρά, τα υπόλοιπα είναι ξένα κλειδιά των άλλων οντοτήτων που αναφέρθηκαν προηγουμένως.

3.1.3.12 Πίνακας Πρωτογενών Δεδομένων (Data Table)

Ο πίνακας Πρωτογενών Δεδομένων έχει τα παρακάτω χαρακτηριστικά μαζί με τους τύπους που έχουν οριστεί:

- Series_ID: INTEGER
- Data_Date: Date
- Status: CHAR(1)
- Value: FLOAT

Το πρώτο είναι ένα ακέραιο αναγνωριστικό κλειδί του πίνακα, το Series_ID είναι ξένο κλειδί της σχέσης Χρονοσειράς τα οποία και τα δύο μαζί σχηματίζουν το πρωτεύον κλειδί της οντότητας. Το Value είναι η δεκαδική τιμή της μέτρησης και το Status δηλώνει με ένα λατινικό χαρακτήρα την κατάσταση των δεδομένων. Πιθανές τιμές για το Data_Status είναι C: Checked, U: Unchecked, I: Invalid, W:Wrong κλπ.

3.1.3.13 Πίνακας Επεξεργασμένων Δεδομένων (Data Table)

Ο πίνακας Επεξεργασμένων Δεδομένων έχει τα παρακάτω χαρακτηριστικά μαζί με τους τύπους που έχουν οριστεί:

- Series_ID: INTEGER
- Data_Date: Date
- Status: CHAR(1)
- Value: FLOAT

Τα πεδία του περιγράφηκαν στον προηγούμενο όμοιο του πίνακα.

3.1.3.14 Πίνακας Σύνθετων Δεδομένων (Data Table)

Ο πίνακας Σύνθετων Δεδομένων έχει τα παρακάτω χαρακτηριστικά μαζί με τους τύπους που έχουν οριστεί:

- Series_ID: INTEGER
- Data_Date: Date
- Status: CHAR(1)
- Value: FLOAT

Τα πεδία του περιγράφηκαν αναλυτικά στον πρώτο από τους δύο προηγούμενους ίδιους πίνακες.

3.2 Περιγραφή αρχιτεκτονικής του προγράμματος διαχείρισης Αίολος

3.2.1 Εισαγωγή

Σε αυτή την παράγραφο, θα δοθούν όλες οι βασικές ενότητες που θα πρέπει να έχει το πρόγραμμα διαχείρισης και παρουσίασης αποτελεσμάτων της βάσης δεδομένων.

3.2.2 Εμφάνιση εγγραφών

Θα πρέπει να δίνεται η δυνατότητα στον χρήστη του προγράμματος να μπορεί να δει τα περιεχόμενα οποιουδήποτε πίνακα της βάσης δεδομένων. Σκόπιμο κρίνεται να παίρνει όλα τα πεδία του πίνακα που θέλει ο χρήστης να δει και να του τα εμφανίζει, έτσι ώστε να του δίνεται η ευκαιρία να επιλέξει αυτός τα πεδία που θέλει να εμφανιστούν. Αυτό μπορεί να γίνει με την παρακάτω SQL εντολή:

```
SELECT column_name FROM all_tab_columns WHERE table_name='MY_TABLE'
```

Αυτή η εντολή θα επιστρέψει όλες τις στήλες που έχει ο πίνακας MY_TABLE. Το πρόγραμμα στην συνέχεια θα πρέπει να τις παρουσιάζει και να εκτελεί το αντίστοιχο select στις στήλες που επέλεξε ο χρήστης:

```
SELECT columns FROM my_table
```

όπου το columns είναι η λίστα με της στήλες που επέλεξε ο χρήστης να δει, χωρισμένες με κόμμα.

Σε αυτή την λειτουργία, θα πρέπει να ενταχθεί και η δυνατότητα να δει ο χρήστης όλους τους πίνακες που είναι διαθέσιμη από την βάση. Αυτό γίνεται με την εκτέλεση της παρακάτω εντολής:

```
SELECT * FROM tab
```

όπου η προβολή tab, κρατάει τα ονόματα όλων των πινάκων που έχουν δημιουργηθεί από τον τρέχον χρήστη.

3.2.3 Εισαγωγή μίας εγγραφής

Για την εισαγωγή μίας εγγραφής στη βάση, θα χρειαστεί ξανά να επιλεγθούν όλες οι στήλες που έχει ο συγκεκριμένος πίνακας που θέλει ο χρήστης να κάνει την εισαγωγή:

```
SELECT column_name FROM all_tab_columns WHERE table_name='MY_TABLE'
```

Μετά για την κάθε στήλη που μας επέστρεψε η παραπάνω ερώτηση, με την βοήθεια ενός βρόχου ζητάμε από τον χρήστη να προσθέσει μία τιμή για το καθένα. Όταν ολοκληρωθεί η εισαγωγή τιμών για όλες τις στήλες από το χρήστη, τότε εκτελείται η SQL εντολή για εισαγωγή των δεδομένων στη βάση:

```
INSERT INTO my_table(columns) VALUES(userinput)
```

με το παραπάνω query, εισάγονται οι τιμές που έδωσε σαν είσοδο ο χρήστης που είναι στο userinput, χωρισμένες με κόμμα και με μονά εισαγωγικά όταν είναι συμβολοσειρά και όχι ακέραιος, στον πίνακα my_table που είχε δηλώσει αρχικά.

3.2.4 Διαγραφή μίας εγγραφής

Εκτός από εισαγωγή, θα πρέπει να δίνεται και η δυνατότητα της διαγραφής μίας συγκεκριμένης εγγραφής. Αυτή μπορεί να χωριστεί σε δύο μέρη. Μπορεί ο χρήστης να μην θέλει να διαγράψει μία συγκεκριμένη εγγραφή, αλλά όλες. Έτσι στην αρχή θα πρέπει να ρωτάει τον χρήστη αν θέλει να τις διαγράψει όλες. Αν ναι, τότε θα εκτελεί την παρακάτω ερώτηση που σβήνει όλες τις εγγραφές:

```
DELETE my_table
```

Επειδή αυτή η ερώτηση δεν θα είναι και η πιο συνηθισμένη, θα πρέπει να έχει ρυθμιστεί η εξορισμού απάντηση σε αυτή την ερώτηση να είναι το όχι. Αν ο χρήστης δεν θέλει να διαγράψει όλες τις εγγραφές, τότε θα του εμφανίζεται το κλειδί του πίνακα και θα περιμένει είσοδο από τον χρήστη για να επιλέξει ποια εγγραφή θέλει να διαγράψει. Όταν την επιλέξει τότε θα εκτελείται η εντολή διαγραφής:

```
DELETE my_table WHERE column=my_value
```

όπου το column είναι η στήλη με την οποία θα γίνει η αναζήτηση για να διαγραφούν οι εγγραφές που έχουν την τιμή που έδωσε ο χρήστης my_value.

3.2.5 Ενημέρωση μίας εγγραφής

Η ενημέρωση μίας εγγραφής θα πρέπει να υποστηρίζεται από το πρόγραμμα διαχείρισης, μιας και είναι αρκετά σημαντική η ανάγκη να διορθωθεί μία εγγραφή που έχει κάποιο σφάλμα. Έτσι ο χρήστης δίνει και εδώ την τιμή για μία στήλη που θέλει να ενημερώσει. Επιστρέφονται όλες οι στήλες και οι τιμές τους που είναι αποθηκευμένες στη βάση και παρουσιάζονται μία μία στον χρήστη για να τους αλλάξει τιμή. Αν ο χρήστης δεν επιθυμεί να αλλάξει την τιμή που θα πρέπει να φαίνεται στο πλάι, τότε με το πάτημα του enter, θα πρέπει η συγκεκριμένη στήλη να παραμείνει στην τιμή αυτή. Όταν ολοκληρωθεί η διαδικασία θα εκτελεστεί η ακόλουθη εντολή ενημέρωσης προς τη βάση δεδομένων:

```
UPDATE my_table SET updated_pair_values
```


όπου το `updated_pair_values` είναι μία λίστα, χωριζόμενη από κόμμα, ζευγαριών στήλης ίσον με τη νέα τιμή που έδωσε ο χρήστης.

3.2.6 Εισαγωγή δεδομένων από αρχείο

Στην σχεδίαση που θα πρέπει να έχει το διαχειριστικό πρόγραμμα, δεν θα πρέπει να ξεχνάμε και την βασική λειτουργία της εισαγωγής δεδομένων από αρχείο. Τα δεδομένα βρίσκονται σε αρχείο ASCII και διαχωρίζονται οι στήλες με ερωτηματικό ή κενό ή οποιονδήποτε άλλο χαρακτήρα. Επειδή η μορφή του αρχείου δεν είναι στάνταρ, για παράδειγμα δεν διαχωρίζονται πάντα οι τιμές με τον ίδιο χαρακτήρα και από αρχείο σε αρχείο υπάρχουν διαφορές στον αριθμό και στη σειρά των χρονοσειρών των αισθητήρων, θα πρέπει να υπάρχει κάποιος τρόπος για να αντιμετωπίσει όλες αυτές τις καταστάσεις. Η λύση σε αυτό είναι να διαβάζεται πρώτα ένα αρχείο περιγραφής του αρχείου δεδομένων. Σε αυτό το αρχείο εκτός από τον διαχωριστή των στηλών και την περιγραφή της σειράς των χρονοσειρών, μπορούν να μπου και άλλες πληροφορίες, όπως είναι ο πίνακας στον οποίο θα φορτωθούν αυτά τα δεδομένα, καθώς και τα πεδία του, ακόμα και την εντολή που θα πραγματοποιήσει με αυτά τα δεδομένα. Θα μπορεί να τα εισάγει, να τα διαγράψει ή να τα ενημερώσει με βάση αυτά που είναι στο αρχείο και αυτά που έχει η βάση.

Η λειτουργία που θα έχει είναι απλή. Θα διαβάσει μία γραμμή από το αρχείο δεδομένων και θα ξεχωρίζει τις στήλες και θα τις αναθέτει στα πεδία που πρέπει για να σχηματίζει το κατάλληλο `insert query`. Αν η εισαγωγή επιστρέψει λάθος, επειδή η εγγραφή υπάρχει ήδη στην βάση, θα πρέπει να υπάρχει επιλογή που θα αναγκάζει την εισαγωγή της εγγραφής από το αρχείο σβήνοντας αυτή που υπάρχει στη βάση ή να επιστρέψει και να τερματίσει με ένα μήνυμα λάθους. Για διαγραφή, απλώς θα διαβάσει τα πεδία και θα στέλνει το αντίστοιχο `delete query` που θα διαγράψει την συγκεκριμένη εγγραφή στην βάση, σύμφωνα με τις τιμές που διάβασε από το αρχείο. Για την ενημέρωση θα σχηματίζει αντί για `insert query`, το αντίστοιχο `update`.

3.2.7 Έλεγχος δεδομένων

Τα δεδομένα που φτάνουν από τον μετεωρολογικό σταθμό, υπεισέρχονται πολλές φορές σε λάθος είτε μετάδοσης είτε λάθος κατά την καταγραφή της τιμής του οργάνου από τον `logger`. Έτσι κρίνεται σκόπιμο να υπάρχει υποστήριξη για έλεγχο της ορθότητας των δεδομένων. Οι μέθοδοι που ακολουθούνται είναι:

- α) Έλεγχος ακραίων τιμών
- β) Έλεγχος χρονικής συνέπειας

Ο έλεγχος ακραίων τιμών είναι η εφαρμογή ενός διαστήματος τιμών που μπορεί να πάρει η μετεωρολογική μεταβλητή που καθορίζεται από μία ελάχιστη και μία μέγιστη. Αν η τιμή δεν είναι μέσα σε αυτό το διάστημα, τότε απορρίπτεται ως σφάλμα μετρήσεως. Αν η τιμή της μετρούμενης μεταβλητής περάσει αυτόν τον έλεγχο, τότε εφαρμόζεται και ο έλεγχος της χρονικής συνέπειας. Σε αυτόν τον έλεγχο, υπολογίζεται η απόλυτη διαφορά της τιμής που μετρήθηκε, σε σχέση με την προηγούμενή της. Αν η διαφορά αυτή είναι μεγαλύτερη από την ορισθείσα, τότε η τιμή απορρίπτεται.

Στην περίπτωση του Αιόλου, χρησιμοποιείται ο πρώτος και ο δεύτερος έλεγχος για να γίνει διόρθωση λανθασμένων τιμών των δεδομένων. Οι ακραίες τιμές και η μέγιστη διαφορά έχουν δοθεί στις περισσότερες σύμφωνα με τις προδιαγραφές του World Meteorological Organisation (WMO), ενώ οι μερικές έχουν διαμορφωθεί σύμφωνα με τις κλιματολογικές συνθήκες που επικρατούν στην δικιά μας περιοχή:

- Θερμοκρασία (°C): ελάχιστη -30, μέγιστη 60, απόκλιση 20
- Σχετική υγρασία (%): ελάχιστη 0, μέγιστη 100, απόκλιση 30
- Ταχύτητα Ανέμου (m/s): ελάχιστη 0, μέγιστη 60, απόκλιση 30

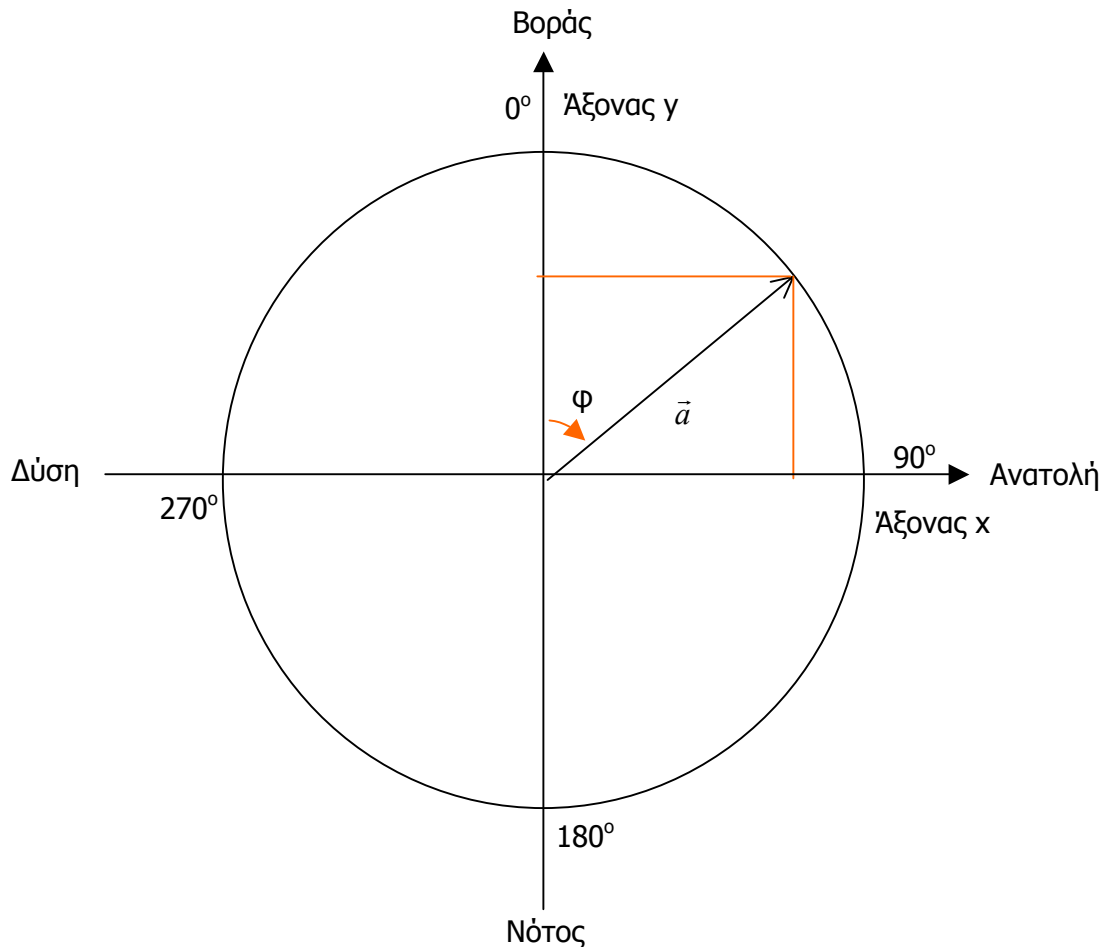
- Διεύθυνση Ανέμου (°): ελάχιστη 0, μέγιστη 360, απόκλιση 360
- Βαρομετρική πίεση (hPa): ελάχιστη 890, μέγιστη 1060, απόκλιση 25
- Βροχόπτωση (mm): ελάχιστη 0, μέγιστη 50, απόκλιση 50
- Ηλιακή ακτινοβολία (W/m^2): ελάχιστη 0, μέγιστη 1500, απόκλιση 1500
- Ολική ηλ. ακτινοβολία ($umol$): ελάχιστη 0, μέγιστη 3000, απόκλιση 3000
- Ηλιοφάνεια (min): ελάχιστη 0, μέγιστη 10, απόκλιση 10

3.2.8 Μετατροπή δεδομένων

Τα διορθωμένα δεδομένα πλέον, είναι σε δεκάλεπτες χρονοσειρές. Αυτές μπορούμε να τις μετατρέψουμε εύκολα σε ωριαίες, τις ωριαίες σε ημερήσιες, τις ημερήσιες σε μηνιαίες και τις μηνιαίες σε ετήσιες. Για την μετατροπή, παίρνουμε τον μέσο όρο, το άθροισμα ή το μέγιστο από την προηγούμενη χρονοσειρά με το αμέσως μικρότερο χρονικό βήμα, ανάλογα με την μετεωρολογική μεταβλητή. Παρακάτω δίνεται η πράξη που γίνεται για την μετατροπή των χρονοσειρών.

- Θερμοκρασία (°C): μέσος όρος
- Σχετική υγρασία (%): μέσος όρος
- Ταχύτητα ανέμου (m/s): μέσος όρος
- Ριπή ανέμου: μέγιστο
- Διεύθυνση ανέμου (°): υπολογισμός γωνίας της συνισταμένης των διανυσμάτων
- Βαρομετρική πίεση (hPa): μέσος όρος
- Βροχόπτωση (mm): άθροισμα
- Ηλιακή ακτινοβολία (W/m^2): μέσος όρος
- Ολική ηλ. ακτινοβολία ($umol$): μέσος όρος
- Διάχυτη ηλ. ακτινοβολία ($umol$): μέσος όρος
- Ηλιοφάνεια (min): άθροισμα

Τα χρονικά διαστήματα για τα οποία προκύπτουν οι αντίστοιχες χρονοσειρές, είναι τα ίδια με αυτά που περιγράφονται στην παράγραφο 1.1.6. Για την διεύθυνση του ανέμου υπολογίζεται το παρακάτω διάνυσμα:



Στο παραπάνω σχήμα φαίνεται η γωνία για την διεύθυνση ανέμου που μετράει το όργανο για μία συγκεκριμένη χρονική στιγμή στον τριγωνομετρικό κύκλο. Για τον υπολογισμό της συνισταμένης των διανυσμάτων που έχουν καταγραφή στο σύνολο του χρονικού διαστήματος που θέλουμε να κάνουμε την μετατροπή της χρονοσειράς, υπολογίζουμε ως εξής.

Έστω φ η διεύθυνση ανέμου και v η ταχύτητα. Τότε τα φ και v ουσιαστικά είναι οι πολικές συντεταγμένες ενός διανύσματος. Τις μετατρέπουμε σε καρτεσιανές ως εξής:

$$\begin{aligned}x &= v \sin \varphi \\y &= v \cos \varphi\end{aligned}$$

Οπότε αν θέλουμε να βγάλουμε μέση ωριαία τιμή, παίρνουμε τα 6 δεκάλεπτα διανύσματα και βγάζουμε το άθροισμά τους:

$$\begin{aligned}X &= x_1 + x_2 + \dots + x_6 \\Y &= y_1 + y_2 + \dots + y_6\end{aligned}$$

Στη συνέχεια πρέπει να μετατρέψουμε τις καρτεσιανές συντεταγμένες του αθροίσματος σε πολικές. Το μήκος το βρίσκουμε εύκολα με τετραγωνική ρίζα του $X^2 + Y^2$, αλλά δεν μας ενδιαφέρει, μόνο η γωνία μας ενδιαφέρει. Η γωνία βρίσκεται ως εξής:

Αν το $Y=0$ τότε $\varphi=90^\circ$, αλλιώς $\varphi=\arctan(|X/Y|)/180\pi$. Τώρα αν το $Y \geq 0$ και $X < 0$ είναι $\varphi=360-\varphi$. Αν $Y < 0$ και $X \geq 0$ τότε $\varphi=180-\varphi$ και τέλος αν $Y < 0$ και $X < 0$ είναι $\varphi=180+\varphi$. Το αποτέλεσμα το στρογγυλοποιούμε και το δίνουμε, μόνο στην περίπτωση που δεν είναι 360. Αν είναι 360 το κάνουμε 0.

3.2.9 Απευθείας εκτέλεση εντολής SQL

Το πρόγραμμα θα προσφέρει βασικές λειτουργίες για εμφάνιση, εισαγωγή, ενημέρωση και διαγραφή εγγραφών από τη βάση δεδομένων. Όμως εκτός από αυτές και για να είναι πιο ευέλικτο, θα πρέπει να προσφέρει και την δυνατότητα στον χρήστη να εκτελεί απευθείας εντολές SQL στην βάση, που δεν καλύπτονται από τις επιλογές που υπάρχουν ήδη.

Έτσι θα πρέπει να δέχεται μία συμβολοσειρά από τον χρήστη, που θα έχει την εντολή που θέλει να εκτελεσθεί στη βάση και θα την μεταβιβάζει. Αν η εντολή ξεκινάει από select, θα πρέπει να μεριμνήσει για να εκτυπώσει τα αποτελέσματα που θα επιστραφούν από την εντολή αυτή.

3.2.10 Παραγωγή γραφημάτων

Τα δεδομένα που έχει η βάση, θα πρέπει να παρουσιάζονται και με τη μορφή διαγραμμάτων για τις τελευταίες 24 ώρες. Αυτό θα πρέπει να το κάνει ο Αίολος με την κατάλληλη επιλογή. Τα διαγράμματα αυτά θα έχουν στον κατακόρυφο άξονα την μετεωρολογική μεταβλητή και στον οριζόντιο άξονα τον χρόνο.

Σε μεταβλητές που έχει νόημα, θα πρέπει να φαίνεται το μέγιστο, το ελάχιστο και η μέση τιμή. Για τα μέγιστα και τα ελάχιστα θα πρέπει να σημειώνεται και η ώρα που εμφανίστηκαν αυτά τα ακρότατα. Επίσης θα πρέπει να μπει και η ημερομηνία και η ώρα που ξεκινάει και τελειώνει ο χρόνος, έτσι ώστε ανημέρωτα διαγράμματα να είναι εύκολο να εντοπιστούν.

3.2.11 Επιλογές του χρήστη

Εκτός από αυτές τις βασικές λειτουργίες που θα πρέπει να έχει, το πρόγραμμα για να είναι ευέλικτο θα πρέπει να δίνει και πολλές επιλογές στον χρήστη, να αλλάζει τις εξορισμού τιμές που έχει. Ένα αρχείο παραμετροποίησης θα μπορούσε να κρατήσει μόνιμα τις επιλογές που επιθυμεί ο χρήστης, με σκοπό να μην χρειάζεται να τις βάζει κάθε φορά σαν παραμέτρους στο πρόγραμμα.

3.2.11.1 Εκτύπωση ή απόκρυψη μηνυμάτων ελέγχου

Θα πρέπει να δίνεται στον χρήστη η δυνατότητα να επιλέγει αν θέλει το πρόγραμμα να του εκτυπώνει μηνύματα για το τι κάνει κάθε φορά ή να μην εκτυπώνει παρά μόνο τα βασικά. Αυτό είναι αρκετά χρήσιμο, ιδιαίτερα όταν κάτι πάει λάθος και θα πρέπει να το εντοπίσεις. Τα μηνύματα λάθους, προφανώς θα πρέπει να εκτυπώνονται κάθε φορά, άσχετα και αν ο χρήστης έχει επιλέξει να μην του εκτυπώνονται καθόλου μηνύματα.

3.2.11.2 Σύνδεση σε διαφορετική βάση ή με διαφορετικό κωδικό

Η σύνδεση στη βάση, θα γίνεται αυτόματα από μία εξορισμού τιμή στο αρχείο παραμετροποίησης ή μέσα στο ίδιο το πρόγραμμα. Ο χρήστης όμως, μπορεί να θελήσει να συνδεθεί σε διαφορετική βάση ή να συνδεθεί στην ίδια με διαφορετικό κωδικό. Αυτό μπορεί να γίνει αν το πρόγραμμα έχει επιλογή για να δώσει ο χρήστης το όνομα της βάσης που θέλει να συνδεθεί, καθώς και τον κωδικό με το σύνθημά του.

3.2.11.3 Διάβασμα από εναλλακτικό αρχείο παραμετροποίησης

Το γενικό αρχείο παραμετροποίησης του προγράμματος, καθώς και το αρχείο παραμετροποίησης των προς εισαγωγή, διαγραφή ή ενημέρωση δεδομένων από αρχείο, θα έχει μία συγκεκριμένη θέση. Ο χρήστης όμως μπορεί να θελήσει να διαβάσει το πρόγραμμα ένα εναλλακτικό αρχείο παραμετροποίησης ή απλώς να θέλει το αρχείο αυτό να βρίσκεται σε διαφορετική τοποθεσία. Έτσι δύο επιλογές για να δώσει εναλλακτική τοποθεσία για το που είναι το αρχείο παραμετροποίησης κρίνονται αναγκαίες.

Ιδιαίτερα για το αρχείο παραμετροποίησης του αρχείου δεδομένων, θα πρέπει να δίνεται και η επιλογή για να εκτελεστεί διαφορετική εντολή εισαγωγής, διαγραφής ή ενημέρωσης, από αυτή που περιγράφεται σε αυτό.

3.2.11.4 Αυτόματη αποθήκευση ή απόρριψη των αλλαγών

Κάθε φορά που θα γίνεται μία αλλαγή στη βάση δεδομένων, το πρόγραμμα θα ρωτάει στο τέλος τον χρήστη, αν επιθυμεί να σώσει τις αλλαγές. Μία χρήσιμη επιλογή θα ήταν να σώζει ή να μην σώζει τις αλλαγές που γίνονται στην βάση, αυτόματα, χωρίς να ρωτάει τον χρήστη για το τι πρέπει να κάνει.

3.2.11.5 Σύντομη τεκμηρίωση προγράμματος

Το πρόγραμμα θα πρέπει να διαθέτει και την βασική επιλογή για να εκτυπώνει ένα σύντομο μήνυμα που θα περιγράφει τις λειτουργίες και τις δυνατότητες που έχει.

3.3 Περιγραφή αρχιτεκτονικής του web interface

Για την παρουσίαση, θα χρησιμοποιηθεί το web interface. Οι σελίδες που θα δημιουργηθούν, θα καλύπτουν τις παρακάτω κατηγορίες. Οι κατηγορίες αυτές θα είναι σε ένα μενού στην κεντρική σελίδα.

3.3.1 Ο καιρός τώρα

Θα υπάρχει μία επιλογή που θα παραπέμπει στον καιρό όπως έχει αυτή την στιγμή. Εκεί θα φαίνονται η τελευταία μέτρηση του αισθητήρα για την αντίστοιχη μετεωρολογική μεταβλητή.

3.3.2 Στατιστικά

Αυτό θα παραπέμπει στα στατιστικά του τελευταίου εικοσιτετράωρου. Θα φαίνεται η τελευταία τιμή της κάθε μεταβλητής, καθώς και τα ακρότατα (μέγιστο και ελάχιστο) και η μέση τιμή για τις τελευταίες 24 ώρες.

3.3.3 Διαγράμματα

Είναι η κατηγορία που θα έχει όλα τα διαγράμματα των μετεωρολογικών μεταβλητών από τους αντίστοιχους αισθητήρες τους. Εκεί θα φαίνονται όλα τα διαγράμματα σε μικρότερες διαστάσεις και με ένα κλικ επάνω σε κάποιο διάγραμμα, ο χρήστης θα πηγαίνει σε άλλο μενού που θα έχει όλες τις μεταβλητές και το διάγραμμα που επέλεξε στις κανονικές του διαστάσεις. Το μενού των μεταβλητών θα έχει τις παρακάτω κατηγορίες.

3.3.3.1 Θερμοκρασία

Το μενού της θερμοκρασίας θα παρουσιάζει το διάγραμμα της θερμοκρασίας. Σε αυτό εκτός από την καμπύλη, θα δίνονται με διακεκομμένη οριζόντια γραμμή το ύψος του μέγιστου, του ελάχιστου και της μέσης τιμής. Με κατακόρυφη διακεκομμένη γραμμή, θα φαίνεται η ακριβής χρονική στιγμή που είχαμε το ακρότατο. Στον κατακόρυφο άξονα θα είναι οι τιμές της θερμοκρασίας σε βαθμούς Κελσίου και στον οριζόντιο ο χρόνος.

3.3.3.2 Βαρομετρική πίεση

Εδώ δίνεται το διάγραμμα της βαρομετρικής πίεσης. Στον κατακόρυφο άξονα είναι η πίεση σε hPa και στον οριζόντιο ο χρόνος σε λεπτά. Ισχύουν και εδώ οι διακεκομμένες γραμμές που δείχνουν τα ακρότατα και τη μέση τιμή.

3.3.3.3 Σχετική υγρασία

Το διάγραμμα της σχετικής υγρασίας δίνεται σε αυτή την ομώνυμη επιλογή του μενού. Φαίνονται και σε αυτό το διάγραμμα τα σημεία που είχε την ανώτερη και την κατώτερη τιμή με τις αντίστοιχες ώρες που συνέβηκαν αυτές, καθώς και η μέση τιμή. Στον κατακόρυφο άξονα θα είναι η υγρασία επί τοις εκατό και στον οριζόντιο ο χρόνος σε λεπτά.

3.3.3.4 Βροχόπτωση

Στο διάγραμμα της βροχόπτωσης δεν θα υπάρχουν ελάχιστο και μέση τιμή, μιας και δεν έχουν νόημα. Θα υπάρχει μόνο η μέγιστη τιμή, για να δείξει το δεκάλεπτο που σημειώθηκε το μεγαλύτερο ύψος βροχής. Στον κατακόρυφο άξονα θα είναι η βροχόπτωση σε mm και στον οριζόντιο ο χρόνος σε λεπτά.

3.3.3.5 Ηλιακή ακτινοβολία

Σε αυτό το διάγραμμα θα δίνεται η ηλιακή ακτινοβολία σε σχέση με τον χρόνο. Ο κατακόρυφος άξονας θα μετράει την ακτινοβολία σε W/m^2 και ο οριζόντιος τον χρόνο σε λεπτά. Με διακεκομμένες γραμμές θα υπάρχουν το μέγιστο, ελάχιστο και η μέση τιμή της ακτινοβολίας για τις τελευταίες 24 ώρες.

3.3.3.6 Διάχυτη ηλιακή ακτινοβολία

Διάγραμμα που θα δίνει στον κατακόρυφο άξονα την διάχυτη ηλιακή ακτινοβολία σε $umol$ και στον οριζόντιο το χρόνο σε λεπτά. Μέση τιμή και ακρότατα φαίνονται με διακεκομμένη γραμμή.

3.3.3.7 Ολική ηλιακή ακτινοβολία

Η ολική ηλιακή ακτινοβολία θα δίνεται και αυτή με τις ακραίες και μέση τιμή. Στον κατακόρυφο άξονα μετράμε την ολική ακτινοβολία σε $umol$ και στον οριζόντιο τον χρόνο σε λεπτά.

3.3.3.8 Ηλιοφάνεια

Για το διάγραμμα της ηλιοφάνειας, έχουμε στον κατακόρυφο άξονα την ηλιοφάνεια σε λεπτά και στον οριζόντιο τον χρόνο σε λεπτά. Το ελάχιστο, το μέγιστο και η μέση τιμή δίνονται με διακεκομμένες γραμμές.

3.3.3.9 Μέση ταχύτητα ανέμου

Η μέση ταχύτητα ανέμου καταγράφεται στον κατακόρυφο άξονα σε m/s και στον οριζόντιο έχουμε τον χρόνο. Με διακεκομμένες φαίνονται η μέγιστη τιμή, η ελάχιστη τιμή και η μέση τιμή.

3.3.3.10 Ριπή ανέμου

Σε αυτό το διάγραμμα φαίνεται η ριπή του ανέμου μαζί με τις ακραίες τιμές και την μέση. Στον κατακόρυφο άξονα δίνεται η ριπή του ανέμου σε m/s και στον οριζόντιο ο χρόνος.

3.3.3.11 Διεύθυνση ανέμου

Η διεύθυνση του ανέμου δίνεται στο διάγραμμα ως συνάρτηση του χρόνο και της μέγιστης τιμής της. Μετράμε στον κατακόρυφο άξονα την διεύθυνση σε μοίρες από 0 έως 360 και στον οριζόντιο τον χρόνο.

3.3.4 Ιστορικά δεδομένα

Στην επιλογή αυτή θα δίνεται η ευκαιρία στον χρήστη να ανακτήσει δεδομένα από το ιστορικό της βάσης δεδομένων. Η ανάκτηση θα γίνεται σε 5 βήματα.

3.3.4.1 Μεταβλητές

Σε αυτό το βήμα θα ζητείται από τον χρήστη να προσδιορίσει τις μετεωρολογικές μεταβλητές από τις οποίες θα πάρει δεδομένα. Αφού τις επιλέξει, θα συνεχίσει παρακάτω για να προσδιορίσει το χρονικό βήμα που θέλει να έχουν αυτές. Τέλος θα προσδιορίσει το χρονικό διάστημα που επιθυμεί. Σε όλα τα παραπάνω οι εξορισμού τιμές θα είναι στο χρονικό βήμα το ωριαίο και στο χρονικό διάστημα η ελάχιστη και η μέγιστη ημερομηνία για την οποία υπάρχουν εγγραφές μέσα στη βάση. Στις μεταβλητές δεν θα είναι καμία προεπιλεγμένη.

3.3.4.2 Αισθητήρες

Το δεύτερο βήμα θα είναι να επιλέξει τους αισθητήρες από τους οποίους θέλει να πάρει δεδομένα. Αυτό το βήμα συνδέεται άμεσα με τις επιλογές που έκανε στο προηγούμενο. Δηλαδή θα του εμφανιστούν οι διαθέσιμοι αισθητήρες μόνο για τις μεταβλητές που επέλεξε. Για κάθε μεταβλητή, υπάρχει χρονοσειρά από τουλάχιστον έναν ή περισσότερους αισθητήρες.

Στον κάθε αισθητήρα που εμφανίζεται, θα υπάρχει:

- εταιρία κατασκευής
- μοντέλο

- περίοδος λειτουργίας
- περίοδος ελλείψεων
- ακρίβεια μέτρησης
- παρατηρήσεις

Ο χρήστης θα μπορεί να πάρει δεδομένα από περισσότερους από έναν αισθητήρα για μία συγκεκριμένη μεταβλητή και αυτές οι πληροφορίες είναι σημαντικές για να την επιλογή ή την απόρριψη κάποιου.

3.3.4.3 Μορφή αρχείου

Το αρχείο που θα παραλάβει ο χρήστης είναι σε μορφή ASCII. Τη μορφή του μπορεί να την τροποποιήσει ανάλογα με τις προτιμήσεις του. Έτσι θα μπορεί να επιλέξει τη μορφοποίηση που θα έχει η ημερομηνία και η ώρα, ποιος θα είναι ο διαχωριστής στηλών και τη σειρά που θα εμφανίζονται οι χρονοσειρές των αισθητήρων μέσα στο αρχείο. Οι εξορισμού τιμές είναι για την ημερομηνία το 'YYYY/MM/DD HH24:MI', για τον διαχωριστή στηλών το ';' και για την σειρά των χρονοσειρών των αισθητήρων, η σειρά που είχαν όπως εμφανίστηκαν στο προηγούμενο βήμα. Μία επιπλέον επιλογή για το αρχείο, είναι αν ο χρήστης επιθυμεί να το λάβει με χωρίς συμπίεση, με συμπίεση ZIP ή με συμπίεση GZIP.

3.3.4.4 Άδεια χορήγησης

Σε αυτό το βήμα παρουσιάζονται όλες οι επιλογές που έχει κάνει ο χρήστης. Του ζητούνται προσωπικές πληροφορίες όπως:

- Όνομα
- Επώνυμο
- Ιδιότητα
- Εταιρία
- Διεύθυνση E-mail
- Σκοπός

Μετά την συμπλήρωση των προσωπικών πληροφοριών, δίνεται παρακάτω και η όροι χορήγησης των δεδομένων. Αφού ο χρήστης την διαβάσει προσεχτικά, επιλέγει το κουμπάκι για υποβολή της αίτησής του.

3.3.4.5 Η αίτηση γίνεται δεκτή

Στο τελευταίο βήμα ο χρήστης ειδοποιείται για την επιτυχή υποβολή της αίτησης χορήγησης δεδομένων. Παράλληλα έχει ξεκινήσει μία άλλη διεργασία για την ανάκτηση των δεδομένων που ζήτησε και όταν ολοκληρωθεί ο χρήστης ειδοποιείται με e-mail για το που θα πρέπει να πάει για να παραλάβει τα δεδομένα του.

3.3.5 Γενικές πληροφορίες

Σε αυτή την κατηγορία παρουσιάζονται όλες οι πληροφορίες που αφορούν τον αυτόματο τηλεμετρικό σταθμό. Αυτές περιλαμβάνουν την τοποθεσία, τις συντεταγμένες, τις μονάδες καταγραφής, τους αισθητήρες που διαθέτει, την τροφοδοσία του, την διακριτότητα αποθήκευσης δεδομένων και τέλος τις μετεωρολογικές μεταβλητές που καταγράφει.

3.3.6 Σύνδεσμοι

Σύνδεσμοι με άλλες παρόμοιες σελίδες μετεωρολογικού περιεχομένου, δίνονται σε αυτή την κατηγορία. Χωρίζονται σε sites που κάνουν μετεωρολογικές προγνώσεις, σε παρόμοιες σελίδες και σε συνδέσμους με sites γενικού περιεχομένου πάνω στην μετεωρολογία.

3.3.7 Φωτογραφίες – Βίντεο

Φωτογραφίες από το σταθμό, καθώς και βίντεο, δίνονται σε αυτή την κατηγορία. Υπάρχουν εικόνες από την ανατολική και δυτική άποψη του σταθμού. Κοντινές φωτογραφίες όπου φαίνονται οι αισθητήρες πάνω στον ιστό και μία γενική φωτογραφία της ομάδας του έργου.

3.3.8 Απαντήσεις ερωτημάτων

Συχνές ερωτήσεις που φτάνουν σε εμάς με επικοινωνία e-mail ή με άλλο τρόπο, μπαίνουν σε αυτή την κατηγορία. Εδώ υπάρχουν απαντήσεις για την πολύ συνηθισμένη ερώτηση αν κάνουμε και προγνώσεις, για κάποιον που θέλει να επισκεφθεί την Ελλάδα από το εξωτερικό και δεν ξέρει πώς είναι ο καιρός εδώ, τι άνεμος επικρατεί για όσους θέλουν να πάνε για ιστιοπλοΐα ή wind surfing, αν έχουμε δεδομένα για άλλες περιοχές στην Ελλάδα και τέλος δίνεται εξήγηση για τις πιθανές απότομες διακυμάνσεις της θερμοκρασίας.

3.3.9 Επικοινωνία – Ομάδα έργου

Σε αυτή την κατηγορία παρουσιάζεται η ομάδα του έργου και δίνεται η ηλεκτρονική διεύθυνση για επικοινωνία. Ακολουθούν επίσης και ευχαριστίες για την συμβολή τους σε όλα τα άτομα που ασχολήθηκαν κατά καιρούς με τον σταθμό.

3.3.10 Εξελίξεις

Εδώ δίνονται όλες οι αλλαγές που γίνονται στις ιστοσελίδες. Γενικά οι εξελίξεις που υπάρχουν στον σταθμό και πιθανόν λάθη και παρατηρήσεις για τα δεδομένα, αναλύονται σε αυτή την κατηγορία.

4 ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ

4.1 Πλατφόρμες και προγραμματιστικά εργαλεία

4.1.1 Βάση δεδομένων Μετεό

Η βάση δεδομένων Μετεό είναι ένας Oracle Server 8.1.5 και η πλατφόρμα που εγκαταστάθηκε και τρέχει είναι Microsoft Windows NT Server 4.0 (service pack 6a). Την στιγμή αυτή που γράφεται το κείμενο έχει βγει η Oracle 8.1.7 και όταν είναι διαθέσιμη στην Ελλάδα, θα γίνει αναβάθμιση σε αυτή και υπάρχει σκέψη να αλλάξει η πλατφόρμα σε Microsoft Windows 2000 Server (service pack 1).

Μαζί με την Oracle, μπήκε και ο Oracle Enterprise Manager 2.0.4 που διαθέτει πολλά προγράμματα για διαχείριση της βάσης δεδομένων. Το σχήμα της βάσης υλοποιήθηκε στο πρόγραμμα Platinum ERWin 3.5.2.

4.1.2 Προγράμματα aeolus, dbonline, index.cgi και histdata.cgi

Η υλοποίηση αυτών των προγραμμάτων, επιλέχτηκε να γραφτεί σε γλώσσα προγραμματισμού Perl. Η Perl (Practical Extraction and Report Language) είναι πολύ καλή στον χειρισμό αρχείων και οι κανονικές εκφράσεις (regular expressions) που χρησιμοποιεί, της δίνουν μεγάλη δύναμη για την εξαγωγή και την αναφορά δεδομένων. Συγκεκριμένα αναπτύχθηκαν σε Perl 5.005 (patch level 3) στο λειτουργικό σύστημα Red Hat Linux 6.2 με πυρήνα έκδοσης 2.2.16. Η υλοποίησή τους έγινε σε VIM έκδοση 5.6.11. Την στιγμή αυτή που γράφεται το παρόν κείμενο, έχει βγει το Red Hat Linux 7.0 με πυρήνα έκδοσης 2.4 και την Perl 5.6. Σύντομα μελετάται η αναβάθμιση σε αυτό το λειτουργικό.

4.1.3 Ιστοσελίδες

Οι σελίδες που δημιουργήθηκαν, εξυπηρετούνται από τον Apache Web server έκδοση 1.3.12 στο λειτουργικό σύστημα Red Hat Linux 6.2. Το πρόγραμμα για την κατασκευή τους ήταν ο VIM έκδοση 5.6.11 στο Linux και το πρόγραμμα Macromedia Dream Weaver 3.0 στα Microsoft Windows. Για τα γραφικά, χρησιμοποιήθηκε το Adobe Photoshop 5.0 και το Macromedia Fire Works 3.0.

4.2 Λεπτομέρειες υλοποίησης

4.2.1 Βάση δεδομένων Μετεό

Η βάση δεδομένων δημιουργήθηκε με βάση το παρακάτω ορισμό:

```
CREATE TABLE AEOLUS.Series_Type (
  Type_ID          INTEGER NOT NULL,
  Name             VARCHAR2(50) NOT NULL,
  GR_Name         VARCHAR2(50) NOT NULL,
  Comments        VARCHAR2(256) NULL
);
```

```
ALTER TABLE AEOLUS.Series_Type
  ADD ( PRIMARY KEY (Type_ID) ) ;
```

```
CREATE TABLE AEOLUS.Station_Type (
  Type_ID          INTEGER NOT NULL,
  Name             VARCHAR2(50) NULL,
  GR_Name          VARCHAR2(50) NULL,
  Comments         VARCHAR2(256) NULL
);
```

```
ALTER TABLE AEOLUS.Station_Type
  ADD ( PRIMARY KEY (Type_ID) ) ;
```

```
CREATE TABLE AEOLUS.Station (
  Station_ID       INTEGER NOT NULL,
  Name             VARCHAR2(50) NOT NULL,
  GR_Name          VARCHAR2(50) NOT NULL,
  Position         VARCHAR2(50) NULL,
  Type_ID          INTEGER NULL,
  Comments         VARCHAR2(256) NULL
);
```

```
ALTER TABLE AEOLUS.Station
  ADD ( PRIMARY KEY (Station_ID) ) ;
```

```
CREATE TABLE AEOLUS.Sensor (
  Sensor_ID        INTEGER NOT NULL,
  Name             VARCHAR2(50) NOT NULL,
  GR_Name          VARCHAR2(50) NOT NULL,
  Station_ID       INTEGER NULL,
  Firm             VARCHAR2(50) NULL,
  Model            VARCHAR2(50) NULL,
  Start_Date       DATE NULL,
  End_Date         DATE NULL,
  Status           CHAR(1) NULL,
  Accuracy         VARCHAR2(10) NULL,
  Comments         VARCHAR2(256) NULL,
  GR_Comments      VARCHAR2(256) NULL
);
```

```
ALTER TABLE AEOLUS.Sensor
  ADD ( PRIMARY KEY (Sensor_ID) ) ;
```

```
CREATE TABLE AEOLUS.Variable (
  Variable_ID      INTEGER NOT NULL,
  Name             VARCHAR2(50) NOT NULL,
  GR_Name          VARCHAR2(50) NOT NULL,
  Unit             VARCHAR2(10) NULL,
  Comments         VARCHAR2(256) NULL
);
```

```
ALTER TABLE AEOLUS.Variable
  ADD ( PRIMARY KEY (Variable_ID) ) ;
```

```

CREATE TABLE AEOLUS.Time_Step (
  Step_ID          INTEGER NOT NULL,
  Name             VARCHAR2(50) NOT NULL,
  GR_Name         VARCHAR2(50) NOT NULL,
  Comments        VARCHAR2(256) NULL
);

ALTER TABLE AEOLUS.Time_Step
  ADD ( PRIMARY KEY (Step_ID) ) ;

CREATE TABLE AEOLUS.Time_Series (
  Series_ID       INTEGER NOT NULL,
  Name            VARCHAR2(50) NOT NULL,
  GR_Name        VARCHAR2(50) NOT NULL,
  Variable_ID    INTEGER NULL,
  Step_ID        INTEGER NULL,
  Type_ID        INTEGER NULL,
  Sensor_ID      INTEGER NULL,
  Comments       VARCHAR2(256) NULL
);

ALTER TABLE AEOLUS.Time_Series
  ADD ( PRIMARY KEY (Series_ID) ) ;

CREATE TABLE AEOLUS.Photo (
  Photo_ID        INTEGER NOT NULL,
  Station_ID      INTEGER NOT NULL,
  Name            VARCHAR2(50) NOT NULL,
  GR_Name        VARCHAR2(50) NOT NULL,
  Pathname       VARCHAR2(255) NULL,
  Comments       VARCHAR2(256) NULL
);

ALTER TABLE AEOLUS.Photo
  ADD ( PRIMARY KEY (Photo_ID, Station_ID) ) ;

CREATE TABLE AEOLUS.Video (
  Video_ID       INTEGER NOT NULL,
  Station_ID     INTEGER NOT NULL,
  Name           VARCHAR2(50) NOT NULL,
  GR_Name       VARCHAR2(50) NOT NULL,
  Pathname      VARCHAR2(255) NULL,
  Comments      VARCHAR2(256) NULL
);

ALTER TABLE AEOLUS.Video
  ADD ( PRIMARY KEY (Video_ID, Station_ID) ) ;

CREATE TABLE AEOLUS.Reason (
  Reason_ID     INTEGER NOT NULL,
  Description   VARCHAR2(50) NOT NULL,
  GR_Description VARCHAR2(50) NOT NULL,
  Comments     VARCHAR2(256) NULL
);

```

```

ALTER TABLE AEOLUS.Reason
  ADD ( PRIMARY KEY (Reason_ID) ) ;

CREATE TABLE AEOLUS.Sensor_NotAvail (
  Sensor_ID          INTEGER NOT NULL,
  Variable_ID       INTEGER NOT NULL,
  Start_Date        DATE NOT NULL,
  End_Date          DATE NOT NULL,
  Reason_ID         INTEGER NULL,
  Comments          VARCHAR2(256) NULL
);

ALTER TABLE AEOLUS.Sensor_NotAvail
  ADD ( PRIMARY KEY (Sensor_ID, Variable_ID, Start_Date) ) ;

CREATE TABLE AEOLUS.Data_Raw (
  Series_ID         INTEGER NOT NULL,
  Data_Date        DATE NOT NULL,
  Status           CHAR(1) NULL,
  Value            FLOAT NULL
);

ALTER TABLE AEOLUS.Data_Raw
  ADD ( PRIMARY KEY (Series_ID, Data_Date) ) ;

CREATE TABLE AEOLUS.Data_Proc (
  Series_ID         INTEGER NOT NULL,
  Data_Date        DATE NOT NULL,
  Status           CHAR(1) NULL,
  Value            FLOAT NULL
);

ALTER TABLE AEOLUS.Data_Proc
  ADD ( PRIMARY KEY (Series_ID, Data_Date) ) ;

CREATE TABLE AEOLUS.Data_Synth (
  Series_ID         INTEGER NOT NULL,
  Data_Date        DATE NOT NULL,
  Status           CHAR(1) NULL,
  Value            FLOAT NULL
);

ALTER TABLE AEOLUS.Data_Synth
  ADD ( PRIMARY KEY (Series_ID, Data_Date) ) ;

ALTER TABLE AEOLUS.Station
  ADD ( FOREIGN KEY (Type_ID)
        REFERENCES AEOLUS.Station_Type ) ;

ALTER TABLE AEOLUS.Sensor
  ADD ( FOREIGN KEY (Station_ID)
        REFERENCES AEOLUS.Station ) ;

```

```

ALTER TABLE AEOLUS.Time_Series
  ADD ( FOREIGN KEY (Variable_ID)
        REFERENCES AEOLUS.Variable ) ;

ALTER TABLE AEOLUS.Time_Series
  ADD ( FOREIGN KEY (Step_ID)
        REFERENCES AEOLUS.Time_Step ) ;

ALTER TABLE AEOLUS.Time_Series
  ADD ( FOREIGN KEY (Type_ID)
        REFERENCES AEOLUS.Series_Type ) ;

ALTER TABLE AEOLUS.Time_Series
  ADD ( FOREIGN KEY (Sensor_ID)
        REFERENCES AEOLUS.Sensor ) ;

ALTER TABLE AEOLUS.Photo
  ADD ( FOREIGN KEY (Station_ID)
        REFERENCES AEOLUS.Station ) ;

ALTER TABLE AEOLUS.Video
  ADD ( FOREIGN KEY (Station_ID)
        REFERENCES AEOLUS.Station ) ;

ALTER TABLE AEOLUS.Sensor_NotAvail
  ADD ( FOREIGN KEY (Reason_ID)
        REFERENCES AEOLUS.Reason ) ;

ALTER TABLE AEOLUS.Sensor_NotAvail
  ADD ( FOREIGN KEY (Variable_ID)
        REFERENCES AEOLUS.Variable ) ;

ALTER TABLE AEOLUS.Sensor_NotAvail
  ADD ( FOREIGN KEY (Sensor_ID)
        REFERENCES AEOLUS.Sensor ) ;

ALTER TABLE AEOLUS.Data_Raw
  ADD ( FOREIGN KEY (Series_ID)
        REFERENCES AEOLUS.Time_Series ) ;

ALTER TABLE AEOLUS.Data_Proc
  ADD ( FOREIGN KEY (Series_ID)
        REFERENCES AEOLUS.Time_Series ) ;

ALTER TABLE AEOLUS.Data_Synth
  ADD ( FOREIGN KEY (Series_ID)
        REFERENCES AEOLUS.Time_Series ) ;

```

Με την εκτέλεση αυτών των εντολών στην Oracle παίρνουμε το ζητούμενο σχήμα που είχε αναλυθεί στο προηγούμενο κεφάλαιο και η βάση είναι έτοιμη για να δεχτεί δεδομένα.

4.2.2 Πρόγραμμα Αίολος

Η λεπτομερής υλοποίηση του προγράμματος διαχείρισης αναλύεται στις παρακάτω παραγράφους.

4.2.2.1 Γενικές δηλώσεις

Η αρχικοποίηση του προγράμματος και οι γενικές δηλώσεις που ορίζει είναι:

```
#!/usr/bin/perl
```

Αυτή είναι η πρώτη γραμμή του προγράμματος που δηλώνει στο λειτουργικό ότι αυτό το πρόγραμμα θα πρέπει να εκτελεστεί με τον Perl διερμηνέα που βρίσκεται στην συγκεκριμένη τοποθεσία που ορίζεται.

```
my $VERSION="1.5.7";
```

Η μεταβλητή \$VERSION κρατάει την τρέχουσα έκδοση του προγράμματος. Η έκδοση χωρίζεται σε τρία ψηφία. Το τελευταίο ψηφίο αυξάνει όταν γίνονται διορθώσεις στο πρόγραμμα. Αλλαγές από προσθήκες διαδικασιών που δεν υπήρχαν, αυξάνουν το μεσαίο αριθμό. Τέλος σημαντικές μεγάλες αλλαγές στην δομή του προγράμματος αυξάνουν τον πρώτο αριθμό.

```
use strict;
use DBI;
use Date::Manip;
use IO::Handle;
use GD;
```

Δηλώσεις χρησιμοποίησης δομοστοιχείων (modules) που χρησιμοποιούνται στο πρόγραμμα. Το strict δομοστοιχείο υπαγορεύει στον διερμηνέα της γλώσσας να χρησιμοποιηθεί αυστηρή σύνταξη στις μεταβλητές, στην αναφορές και στις ρουτίνες.

```
# Variable Declaration
my ($dbservice, $data_source, $dblogin, $dbpassword);
my $configfile="/etc/aeolus/aeolus.conf";
my $IMAGEDIR="/home/httpd/html/newmeteo/images";
my ($verbose,$debug);
my %dbattr=( PrintError => 0, RaiseError => 0, AutoCommit => 0 );
my $PROGNAME="aeolus";
my $SERVERNAME=`/bin/hostname -s`;
```

Σε αυτό το κομμάτι ορίζονται οι γενικές μεταβλητές που θα χρησιμοποιηθούν από όλες τις ρουτίνες του προγράμματος. Τα \$dbservice, \$data_source, \$dblogin και \$dbpassword χρησιμεύουν για την σύνδεση στη βάση δεδομένων. Η μεταβλητή \$configfile κρατάει το μονοπάτι του αρχείου παραμετροποίησης του προγράμματος και το \$IMAGEDIR το μονοπάτι που θα φυλαχτούν τα διαγράμματα. Οι γενικές μεταβλητές \$verbose και \$debug, φέρουν την πληροφορία για την αναλυτική εκτύπωση μηνυμάτων από την βάση και από το πρόγραμμα αντίστοιχα. Το hash %dbattr ορίζει επιλογές κατά τη σύνδεση στη βάση και αρχικοποιεί το PrintError (αυτόματη εκτύπωση σφάλματος), το RaiseError (αυτόματη διακοπή προγράμματος στην περίπτωση λάθους) και το AutoCommit (αυτόματο σώσιμο των αλλαγών στη βάση) στο μηδέν. Οι ακόλουθες scalar μεταβλητές \$PROGNAME και \$SERVERNAME κρατούν το όνομα του προγράμματος και του υπολογιστή στον οποίο εκτελείται.

```
my ($dbh, $sth, $command, $tablename, $filename, $sql, $msg);
my (@row, $loadcommand, %table, $fieldnumber, $delimiter,%function,%format);
my ($loadconfig, $forceinsert);
```



```

print "-ac\t\tenable auto commit option\n";
print "-nac\t\tdisable auto commit option\n";
print "-C <config>\tspecify an alternative location for config file\n";
print "-lC <loadconfig>\tspecify an alternative location for load config\n";
print "-d <no>\t\tsetting debug level (valid <no> are from 0 to 5)\n\n";
}

```

Η συγκεκριμένη ρουτίνα όταν καλείται κατά την εκτέλεση της εντολής `help` σαν όρισμα στο πρόγραμμα, το μόνο που κάνει είναι να εκτυπώσει τα παραπάνω μηνύματα που δίνονται σαν όρισμα στην εντολή `print`.

4.2.2.3 Ρουτίνα εκτύπωσης μηνυμάτων `sysmsg`

Για την εκτύπωση των μηνυμάτων στον χρήστη, χρησιμοποιείται η ρουτίνα `sysmsg`:

```

sub sysmsg
{
  if (defined $_[1])
  {
    if (defined $debug)
    {
      if ($debug>= $_[1])
      {
        print $_[0];
      }
    }
    else
    {
      print $_[0];
    }
  }
  else
  {
    print "Usage: sysmsg(<output string>, <print status>)\n";
    print "Print status:\n";
    print "0\t\talways print the output string\n";
    print "n\t\tprint string only in if debug level is greater or equal to n\n";
    exit -1;
  }
}

```

Δέχεται δύο ορίσματα, ένα για την συμβολοσειρά που θα εκτυπώσει και ένα για τον αριθμό του επιπέδου `debug`. Η λειτουργία της είναι η εξής. Αν είναι ορισμένο μέχρι και το δεύτερο όρισμα, τότε ελέγχει αν είναι ορισμένη η μεταβλητή `$debug`. Αν ναι εκτυπώνει το μήνυμα όταν το επίπεδο `debug` είναι μεγαλύτερο ή ίσο από τον αριθμό που έχει περάσει για `print status`. Αν δεν έχει οριστεί επίπεδο `debug` το εκτυπώνει ούτως ή άλλως. Σε περίπτωση που δεν υπάρχουν 2 ορίσματα στην ρουτίνα, τερματίζει με ένα σύντομο μήνυμα χρήσης της.

4.2.2.4 Ρουτίνα εκτύπωσης σφαλμάτων `syserr`

Και για την εκτύπωση των μηνυμάτων σφάλματος, χρησιμοποιείται η ρουτίνα `syserr` για να κάνει συγκεκριμένη διαμόρφωση στην συμβολοσειρά του λάθους που θα εκτυπωθεί στον χρήστη.

```

sub syserr
{
  if (defined $_[1])
  {

```

```

unless ($_[1]==0)
{
    warn "ERROR($_[1]) [$PROGNAME]: $_[0]\n";
    exit $_[1];
}
else
{
    warn "WARNING [$PROGNAME]: $_[0]\n";
}
}
else
{
    print "Usage: syserr(<message string>, <exit status>)\n";
    print "Exit status:\n";
    print "0\t\tprints a warning and continues script execution\n";
    print "n\t\tprints the error message and exits with the nonzero n status\n";
    exit -1;
}
}
}

```

Δέχεται δύο ορίσματα, από τα οποία το ένα είναι το μήνυμα λάθους και το δεύτερο είναι ένας αριθμός χαρακτηριστικός για το σημείο στο οποίο βρέθηκε το λάθος. Αν δεν υπάρχουν αυτά τα δύο ορίσματα, εκτυπώνει το σύντομο μήνυμα χρήσης της και τερματίζει το πρόγραμμα. Στην περίπτωση που ο αριθμός σφάλματος είναι 0, τότε αυτό εκτυπώνεται σαν προειδοποίηση και το πρόγραμμα συνεχίζει την λειτουργία του, ενώ αν είναι μη μηδενικό το σφάλμα εκτυπώνεται σαν λάθος και το πρόγραμμα τερματίζει επιστρέφοντας στο λειτουργικό τον συγκεκριμένο αριθμό σφάλματος.

4.2.2.5 Ρουτίνα προσθήκης μονών εισαγωγικών QuoteStrings

Για να χρησιμοποιηθούν συμβολοσειρές σε εντολές SQL, θα πρέπει να είναι μέσα σε μονά εισαγωγικά. Η ρουτίνα QuoteStrings υλοποιήθηκε για την αυτόματη εισαγωγή των μονών εισαγωγικών στις συμβολοσειρές:

```

# QuoteStrings function. Error level 6

sub QuoteStrings
{
    my (@quoted, $result);
    my @isNumber=DBI::looks_like_number(@_);
    my $i=0;

    while (defined $_[ $i ])
    {
        unless ($isNumber[$i])
        {
            $quoted[$i]=$dbh->quote($_[$i]);
        }
        else
        {
            $quoted[$i]=$_[$i];
        }
        $i++;
    }
    $result=join ' ', @quoted;

    return $result;
}

```

Η ρουτίνα όσο έχει ορίσματα, τα ελέγχει αν είναι αριθμοί και αν είναι τους αποθηκεύει όπως είναι χωρίς εισαγωγικά στον πίνακα @arguments και αν όχι τους βάζει μονά εισαγωγικά και τους αποθηκεύει ξανά στον ίδιο πίνακα. Στο τέλος επιστρέφει το αποτέλεσμα σαν συμβολοσειρά με τις κάθε τιμές διαχωριζόμενες με κόμμα.

4.2.2.6 Ρουτίνα εκτέλεσης ερώτησης RunQuery

Η εκτέλεση των εντολών SQL στη βάση δεδομένων, έχουν τυποποιηθεί σε αυτή την ρουτίνα:

```
# RunQuery function. Error level 13

sub RunQuery
{
  my $err=13;
  my (@arguments,$query,$message,$error);

  @arguments=@_;
  $query=$arguments[0];
  $message=$arguments[1];
  $error=$arguments[2];

  syserr("Usage: RunQuery(<query>,<message>,<error number>)", $err)
  unless ($#arguments==2);

  $sth=$dbh->prepare($query) or
  syserr("Can't prepare $message:\nDB error: $DBI::errstr", $error);
  $sth->execute or
  syserr("Can't execute $message:\nDB error: $DBI::errstr", $error);
  $queries++;
}
```

Σαν ορίσματα δέχεται την ερώτηση που θα εκτελεσθεί στη βάση, ένα μήνυμα που περιγράφει την προς εκτέλεση εντολή και το επίπεδο λάθους. Αν τα ορίσματα δεν είναι τρία, τότε εκτυπώνει ένα σύντομο μήνυμα χρήσης και τερματίζει, αλλιώς προετοιμάζει την ερώτηση και την εκτελεί. Το αποτέλεσμα επιστρέφεται στην γενική μεταβλητή \$sth. Η μετρητής των εντολών που εκτελούνται στη βάση \$queries, αυξάνει κατά ένα.

4.2.2.7 Ρουτίνα διαβάσματος ορισμάτων ReadArguments

Η ρουτίνα αυτή, διαβάζει ένα ένα τα ορίσματα που έχει δώσει ο χρήστης στο πρόγραμμα και ρυθμίζει κατάλληλα διάφορες μεταβλητές:

```
# ReadArguments function. Error level 3

sub ReadArguments
{
  my $i=0;

  while (defined $ARGV[$i])
  {
```

Όσο υπάρχουν ορίσματα, εκτελεί μία από τις παρακάτω συνθήκες.

```
  if ($ARGV[$i] eq "show")
  {
    $command=$ARGV[$i];
```

```

$i++;
if (defined $ARGV[$i])
{
    $tablename=$ARGV[$i];
}
else
{
    syserr("Missing table name for \"show\" command.", 3);
}
$tablename=uc($tablename);
}

```

Αν η εντολή είναι "show", τότε ρυθμίζει την μεταβλητή \$command και αυξάνει τον μετρητή την τρέχουσα θέση των ορισμάτων ϕ κατά ένα. Μετά ελέγχει για την ύπαρξη και δεύτερου ορίσματος, που είναι το όνομα του πίνακα της βάσης που θα δείξει τα περιεχόμενα και ορίζει την μεταβλητή \$tablename.

```

elseif ($ARGV[$i] eq "showall")
{
    $command=$ARGV[$i];
}

```

Αν η εντολή είναι "showall", τότε ρυθμίζει σε αυτή την τιμή την \$command.

```

elseif ($ARGV[$i] eq "add")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        $tablename=$ARGV[$i];
    }
    else
    {
        syserr("Missing table name for \"add\" command.", 3);
    }
    $tablename=uc($tablename);
}
elseif ($ARGV[$i] eq "delete")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        $tablename=$ARGV[$i];
    }
    else
    {
        syserr("Missing table name for \"delete\" command.", 3);
    }
    $tablename=uc($tablename);
}
elseif ($ARGV[$i] eq "update")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        $tablename=$ARGV[$i];
    }
    else
    {
        syserr("Missing table name for \"update\" command.", 3);
    }
}

```

```

    }
    $tablename=uc($tablename);
}

```

Για την περίπτωση των εντολών `add`, `delete` ή `update` ορίζεται η μεταβλητή `$command` σε αυτή τη συμβολοσειρά. Αυξάνεται ο μετρητής της τρέχουσας θέσης των ορισμάτων `$i` και ελέγχεται αν έχει δοθεί και άλλο όρισμα που θα δηλώνει τον πίνακα στον οποίο θα πραγματοποιηθεί η προσθήκη, αφαίρεση ή διαγραφή μίας εγγραφής.

```

elseif ($ARGV[$i] eq "load")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        $filename=$ARGV[$i];
    }
    else
    {
        syserr("Missing file name for \"load\" command.",3);
    }
}

```

Όταν η εντολή είναι `"load"` ρυθμίζεται η `$command`, καθώς και το όνομα του αρχείου που θα πρέπει να συνοδεύει αυτή την εντολή.

```

elseif ($ARGV[$i] eq "make")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        if ("normalised hourly daily monthly yearly"=~/$ARGV[$i]/i)
        {
            $timestep=$ARGV[$i];
        }
        else
        {
            syserr("Wrong time series name \"$ARGV[$i]\" to produce",3);
        }
    }
    else
    {
        syserr("Missing time step to produce in $command command",3);
    }
    $i++;
    if (defined $ARGV[$i])
    {
        $timeseries=$ARGV[$i];
    }
    else
    {
        syserr("Missing time series to produce in $timestep series",3);
    }
}

```

Η εντολή `"make"` ακολουθείται από δύο ορίσματα: το χρονικό βήμα και την χρονοσειρά την οποία θα μετατρέψει ή θα παράγει. Αν λείπει κάποιο από αυτά, τότε εκτυπώνεται το κατάλληλο μήνυμα λάθους και το πρόγραμμα τερματίζει.

```

elseif ($ARGV[$i] eq "execute")
{

```

```

$command=$ARGV[$i];
$i++;
if (defined $ARGV[$i])
{
    $execute=$ARGV[$i];
}
else
{
    syserr("Missing SQL statement to execute in \"execute\" command",3);
}
}

```

Η απευθείας εκτέλεση εντολών στην βάση δεδομένων, επιτυγχάνεται με την εντολή “execute” στο πρόγραμμα. Εδώ ρυθμίζεται η \$command και στην μεταβλητή \$execute φυλάγεται η εντολή που θα σταλθεί απευθείας στη βάση.

```

elseif ($ARGV[$i] eq "generate")
{
    $command=$ARGV[$i];
    $i++;
    if (defined $ARGV[$i])
    {
        if ("ten-minute hourly daily monthly"=~/$ARGV[$i]/i)
        {
            $timestep=$ARGV[$i];
        }
        else
        {
            syserr("Unsupported graph name \"$ARGV[$i]\" to produce",3);
        }
    }
    else
    {
        syserr("Missing time step to produce in $command command",3);
    }
    $i++;
    if (defined $ARGV[$i])
    {
        $timeseries=$ARGV[$i];
    }
    else
    {
        syserr("Missing time series to produce in $timestep graph",3);
    }
}
}

```

Η εντολή “generate” παράγει τα διαγράμματα για το αντίστοιχο χρονικό βήμα που θα προσδιοριστεί και για ποια χρονοσειρά. Αν δεν ακολουθεί την εντολή το χρονικό βήμα ή η χρονοσειρά, τότε εκτυπώνεται το αντίστοιχο μήνυμα λάθους και το πρόγραμμα τερματίζει.

```

elseif ($ARGV[$i] eq "help")
{
    Usage;
    exit 0;
}

```

Στην εντολή “help” το πρόγραμμα εκτελεί την συνάρτηση Usage που δίνει μία σύντομη βοήθεια στην σύνταξη, στις εντολές και τις επιλογές του προγράμματος και τερματίζει με επιτυχία.

```

elseif ($ARGV[$i] eq "-v")
{
    $i++;
}

```

```

if (defined $ARGV[$i])
{
    $verbose=$ARGV[$i];
    $dbattr{PrintError}=( $verbose eq "yes"?1:0);
}
else
{
    syserr("Missing yes or no in \"-v\" option",3);
}
}

```

Η επιλογή "-v" θέτει την παράμετρο PrintError της σύνδεσης με την βάση στο 1, όταν έχει δοθεί θετική απάντηση και 0 όταν έχει δοθεί αρνητική. Η μεταβλητή \$verbose κρατάει την απάντηση που έχει δώσει ο χρήστης.

```

elsif ($ARGV[$i] eq "-s")
{
    $i++;
    if (defined $ARGV[$i])
    {
        $dbservice=$ARGV[$i];
    }
    else
    {
        syserr("Missing database service in \"-s\" option",3);
    }
}

```

Για να προσδιοριστεί διαφορετική βάση δεδομένων από αυτή που περιγράφεται στο αρχείο παραμετροποίησης, ο χρήστης μπορεί να δώσει την επιλογή "-s" να μετά να δώσει το όνομα της βάσης που θέλει να συνδεθεί. Η μεταβλητή \$dbservice ρυθμίζεται αντίστοιχα στην είσοδο που έδωσε ο χρήστης.

```

elsif ($ARGV[$i] eq "-u")
{
    $i++;
    if (defined $ARGV[$i])
    {
        if ($ARGV[$i]=~/\/\/)
        {
            ($dblogin, $dbpassword)=split /\\/\/, $ARGV[$i];
        }
        else
        {
            $dblogin=$ARGV[$i];
        }
    }
    else
    {
        syserr("Missing database username in \"-u\" option",3);
    }
}

```

Εκτός από διαφορετικό όνομα βάσης δεδομένων, μπορεί να προσδιορίσει και διαφορετικό όνομα χρήστη ή και συνθήματος χρήστη για να συνδεθεί με διαφορετικό κωδικό. Αν ο χρήστης έχει δώσει συμβολοσειρά της μορφής «χρήστης/σύνθημα» τότε ρυθμίζεται εκτός από την μεταβλητή \$dblogin και η \$dbpassword. Διαφορετικά ρυθμίζεται μόνο η \$dblogin.

```

elsif ($ARGV[$i] eq "-p")
{
    $i++;

```

```

if (defined $ARGV[$i])
{
    $dbpassword=$ARGV[$i];
}
else
{
    syserr("Missing database user password in \"-p\" option",3);
}
}

```

Το σύνθημα του χρήστη που θα συνδεθεί στη βάση, μπορεί να το δώσει και ξεχωριστά με την επιλογή "-p".

```

elsif ($ARGV[$i] eq "-l")
{
    $i++;
    if (defined $ARGV[$i])
    {
        $loadcommand=$ARGV[$i];
    }
    else
    {
        syserr("Missing load command in \"-l\" option",3);
    }
}

```

Κατά το φόρτωμα ενός αρχείου δεδομένων, η εξορισμού εντολή που θα εκτελέσει με τα δεδομένα περιγράφεται στο αρχείο παραμετροποίησης του συγκεκριμένου αρχείου δεδομένων. Ο χρήστης με την επιλογή "-l" μπορεί να δώσει εναλλακτική εντολή.

```

elsif ($ARGV[$i] eq "-y")
{
    $ASK4SAVE="no";
    $savechanges="yes";
}
elsif ($ARGV[$i] eq "-ny")
{
    $ASK4SAVE="no";
    $savechanges="no";
}

```

Όταν γίνεται κάποια αλλαγή στη βάση, το πρόγραμμα ρωτάει στο τέλος τον χρήστη αν θέλει να σώσει τις αλλαγές. Με τις επιλογές "-y" και "-ny" το πρόγραμμα δεν ρωτάει αν θα σώσει τις αλλαγές και υποθέτει ότι η απάντηση στο αν θα τις σώσει είναι «ναι» και «όχι» αντίστοιχα. Η μεταβλητή \$ASK4SAVE τίθεται στο "no" για να μην γίνει η σχετική ερώτηση στο τέλος και η \$savechanges παίρνει την τιμή ανάλογα με ποια επιλογή χρησιμοποιήθηκε.

```

elsif ($ARGV[$i] eq "-ac")
{
    $autocommit="yes";
}
elsif ($ARGV[$i] eq "-nac")
{
    $autocommit="no";
}

```

Με αυτές τις επιλογές ρυθμίζεται το AutoCommit όρισμα που περνάει στη βάση δεδομένων κατά τη σύνδεση. Η μεταβλητή \$autocommit ρυθμίζεται ανάλογα.

```

elsif ($ARGV[$i] eq "-f")
{

```



```

    $forceinsert="yes";
}

```

Όταν εισάγονται δεδομένα από αρχείο, τότε αν υπάρχει ήδη η αντίστοιχη εγγραφή, το πρόγραμμα τερματίζει με μήνυμα λάθους. Με την επιλογή "-f" το πρόγραμμα θα σβήσει την υπάρχουσα εγγραφή και θα εισάγει την καινούργια.

```

elseif ($ARGV[$i] eq "-d")
{
    $i++;
    if (defined $ARGV[$i])
    {
        $debug=$ARGV[$i];
    }
    else
    {
        syserr("Missing debug number in \"-d\" option",3);
    }
}
}

```

Με την επιλογή "-d" ρυθμίζεται το επίπεδο debug. Το διάστημα είναι από 0 έως 5.

```

elseif ($ARGV[$i] eq "-C")
{
    $i++;
    if (defined $ARGV[$i])
    {
        $configfile=$ARGV[$i];
    }
    else
    {
        syserr("Missing configuration file for \"-C\" option",3);
    }
}
}

```

Το αρχείο παραμετροποίησης του προγράμματος έχει συγκεκριμένη θέση που ορίζεται από την μεταβλητή \$configfile. Ο χρήστης μπορεί με την επιλογή "-C" να δώσει εναλλακτική θέση του αρχείου αυτού.

```

elseif ($ARGV[$i] eq "-lC")
{
    $i++;
    if (defined $ARGV[$i])
    {
        $loadconfig=$ARGV[$i];
    }
    else
    {
        syserr("Missing load configuration file for \"-lC\" option",3);
    }
}
}

```

Για αρχείο παραμετροποίησης του αρχείου δεδομένων διαβάζεται εξορισμού το ίδιο αρχείο, στο ίδιο μονοπάτι με το αρχείο δεδομένων που έχει την κατάληξη ".load". Με την επιλογή "-lC" μπορεί να δοθεί κάποιο διαφορετικό.

```

else
{
    syserr("Command or option \"$ARGV[$i]\" is not supported. Ignoring...",
0);
}
}

```

```

    $i++;
}

```

Αν το όρισμα που δόθηκε δεν είναι κάποιο από αυτά που περιγράφηκαν, τότε το πρόγραμμα το αγνοεί ως λάθος και συνεχίζει εκτυπώνοντας την αντίστοιχη προειδοποίηση.

```

if ($i==0)
{
    syserr("No options specified. Type \"$PROGNAME help\" for a brief help",3);
}
unless (defined $command)
{
    syserr("Undefined command action",3);
}
}

```

Αν δεν έχουν δοθεί καθόλου ορίσματα ή δεν έχει οριστεί κάποια εντολή εκτέλεσης, τότε το πρόγραμμα τερματίζει.

4.2.2.8 Ρουτίνα διαβάσματος αρχείου παραμετροποίησης ReadConfig

Για το διάβασμα του γενικού αρχείου παραμετροποίησης, χρησιμοποιείται η ρουτίνα ReadConfig:

```

# ReadConfig function. Error level 2

sub ReadConfig
{
    my $SERVICE="DBService";
    my $LOGIN="DBLogin";
    my $PASSWORD="DBPassword";
    my $VERBOSE="Verbose";
    my $DEBUG="Debug";
    my $FORCEINSERT="ForceInsert";
    my $SAVECHANGES="SaveChanges";
    my $AUTOCOMMIT="AutoCommit";
}

```

Παραπάνω έχουν οριστεί ως τοπικές μεταβλητές οι λέξεις κλειδιά του αρχείου παραμετροποίησης.

```

my ($keyword,$confservice,$conflogin,$confpassword,$confverbose,$confdebug);
my ($confforceinsert,$confsavechanges,$confautocommit);

```

Το \$keyword είναι μία βοηθητική μεταβλητή που κρατάει την λέξη κλειδί που διαβάστηκε στο αρχείο. Οι μεταβλητές \$conf* χρησιμοποιούνται για να καταχωρήσουν την τιμή της λέξης κλειδί που διαβάστηκε.

```

sysmsg("Reading $configfile configuration file...\n",1);

```

Εκτύπωση μηνύματος για το διάβασμα του αρχείου παραμετροποίησης.

```

if (-e $configfile)
{
    if (-r $configfile)
    {

```

Αν υπάρχει το αρχείο και αν έχει τις κατάλληλες άδειες για διάβασμα, τότε προχωρά στην ανάγνωσή του.

```

    open(CFH,$configfile)

```

```
or syserr("Couldn't read from $configfile:\nSystem error: $!",2);
```

Άνοιγμα του αρχείου για διάβασμα.

```
while (<CFH>)
{
```

Όσο υπάρχουν γραμμές κάνει τα παρακάτω.

```
if ($_ =~ /^#/)
{
    next;
}
```

Αν η γραμμή ξεκινάει από '#', τότε αυτή θεωρείται σχόλιο και αγνοείται.

```
elseif ($_ =~ /\w/)
{
    if ($_ =~ /$SERVICE/i)
    {
        ($keyword, $confservice)=split(/=/, $_);
        chomp $confservice;
        $confservice=~tr/"'//d;
        $dbservice=$confservice unless (defined $dbservice);
    }
}
```

Αν υπάρχει έστω και ένας χαρακτήρας που είναι κανονικός χαρακτήρας και όχι κενό ή αλλαγή γραμμής, ελέγχει να δει σε ποια περίπτωση κλειδιού βρίσκεται. Αν είναι στην περίπτωση που υπάρχει δήλωση για το όνομα της βάσης, τότε χωρίζει την λέξη κλειδί από την τιμή της με βάση το '=' και τις αναθέτει στις μεταβλητές \$keyword και \$confservice. Στην συνέχεια αφαιρείται το τέλος της γραμμής (end of line) και τα διπλά ή μονά εισαγωγικά που μπορεί να έχει. Η γενική μεταβλητή για το όνομα της βάσης \$dbservice παίρνει την τιμή που διαβάστηκε στο όνομα αρχείου, μόνο όταν δεν έχει οριστεί προηγουμένως στο πρόγραμμα από την γραμμή εντολών. Οι παρακάτω λογικές διακλαδώσεις εκτελούν την ίδια λειτουργία και για τα υπόλοιπα ορίσματα του αρχείου παραμετροποίησης.

```
elseif ($_ =~ /$LOGIN/i)
{
    ($keyword, $conflogin)=split /=/, $_;
    chomp $conflogin;
    $conflogin=~tr/"'//d;
    $dblogin=$conflogin unless (defined $dblogin);
}
elseif ($_ =~ /$PASSWORD/i)
{
    ($keyword, $confpassword)=split /=/, $_;
    chomp $confpassword;
    $confpassword=~tr/"'//d;
    $dbpassword=$confpassword unless (defined $dbpassword);
}
elseif ($_ =~ /$VERBOSE/i)
{
    ($keyword, $confverbose)=split /=/, $_;
    chomp $confverbose;
    $confverbose=~tr/"'//d;
    $verbose=$confverbose unless (defined $verbose);
}
elseif ($_ =~ /$DEBUG/i)
{
    ($keyword, $confdebug)=split /=/, $_;
    chomp $confdebug;
```

```

    $confdebug=~tr/"'//d;
    $debug=$confdebug unless (defined $debug);
}
elseif ($_~/FORCEINSERT/i)
{
    ($keyword, $confforceinsert)=split /=/, $_;
    chomp $confforceinsert;
    $confforceinsert=~tr/"'//d;
    $forceinsert=$confforceinsert unless (defined $forceinsert);
}
elseif ($_~/SAVECHANGES/i)
{
    ($keyword,$confsavechanges)=split /=/, $_;
    chomp $confsavechanges;
    $confsavechanges=~tr/"'//d;
    $savechanges=$confsavechanges unless (defined $savechanges);
}
elseif ($_~/AUTOCOMMIT/i)
{
    ($keyword,$confautocommit)=split /=/, $_;
    chomp $confautocommit;
    $confautocommit=~tr/"'//d;
    $autocommit=$confautocommit unless (defined $autocommit);
}
else
{
    syserr("Unsupported configuration line:\n$_",2);
}
}
}

```

Αν ο έλεγχος των λέξεων κλειδιών του αρχείου παραμετροποίησης φτάσει στο τέλος και έχει βρεθεί μία λέξη που δεν ανήκει στο σύνολο αυτό που μπορεί να πάρει το αρχείο αυτό, τότε εκτυπώνεται ένα μήνυμα λάθους στο χρήστη με την συγκεκριμένη γραμμή που βρέθηκε αυτό.

```

sysmsg("Configuration file was read as following:\n",3);
sysmsg("$SERVICE=$confservice ($dbservice)\n",3)
if (defined $confservice);
sysmsg("$LOGIN=$conflogin ($dblogin)\n",3)
if (defined $conflogin);
sysmsg("$PASSWORD=$confpassword ($dbpassword)\n",3)
if (defined $confpassword);
sysmsg("$VERBOSE=$confverbose ($verbose)\n",3)
if (defined $confverbose);
sysmsg("$DEBUG=$confdebug ($debug)\n",3)
if (defined $confdebug);
sysmsg("$FORCEINSERT=$confforceinsert ($forceinsert)\n",3)
if (defined $confforceinsert);
sysmsg("$SAVECHANGES=$confsavechanges ($savechanges)\n",3)
if (defined $confsavechanges);
sysmsg("$AUTOCOMMIT=$confautocommit ($autocommit)\n",3)
if (defined $confautocommit);
}

```

Αν το πρόγραμμα έχει τρέξει σε debug level 3 ή μεγαλύτερο, τότε εκτυπώνεται στον χρήστη οι λέξεις κλειδιά που διαβάστηκαν από το αρχείο, μαζί με τις τιμές τους. Δίπλα σε παρένθεση δίνεται η τελική δήλωση της επιλογής, που αν δεν έχει δοθεί στην γραμμή εντολών, θα είναι η ίδια με αυτή που διαβάστηκε από το αρχείο παραμετροποίησης.

```

else
{
    syserr("Config file $configfile exists, but it's unreadable", 2);
}

```

```

    }
  }
  else
  {
    syserr("Configuration file $configfile does not exist", 2);
  }
  close(CFH);
  $data_source="DBI:Oracle:$dbservice";
}

```

Οι δύο τελευταίες συνθήκες εκτελούνται όταν το αρχείο δεν έχει τις κατάλληλες άδειες για ανάγνωση και όταν δεν υπάρχει. Στο τέλος κλείνεται και ο χειριστής του αρχείου παραμετροποίησης που διαβάστηκε και ορίζεται η μεταβλητή `$data_source` που θα προσδιορίσει στο κυρίως πρόγραμμα τον οδηγό που θα χρησιμοποιήσει για να συνδεθεί στη βάση δεδομένων.

4.2.2.9 Ρουτίνα ανάγνωσης αρχείου παραμετροποίησης δεδομένων ReadLoadConfig

Η παραμετροποίηση ενός αρχείου δεδομένων, γίνεται με την βοήθεια ενός αρχείου που περιγράφει τον πίνακα της βάσης που θα καταλήξουν τα δεδομένα, τον αριθμό των στηλών που περιέχει, την εντολή που θα εκτελέσει, τον διαχωριστή στηλών και την αντιστοιχία των πεδίων του πίνακα με τις τιμές στις στήλες του αρχείου.

```

# ReadLoadConfig function. Error level 7

sub ReadLoadConfig
{
  my $TARGETTABLE="TargetTable";
  my $ACTION="Action";
  my $FIELD="Field";
  my $RANGE="Range";
  my $DELIMITER="Delimiter";
  my $FIELDNUMBER="FieldNumber";
  my $FUNCTION="Function";
  my $FORMAT="Format";

```

Οι μεταβλητές αυτές κρατούν τις λέξεις κλειδιά που μπορούν να εμφανιστούν στο αρχείο παραμετροποίησης.

```

  my ($keyword, $field, $validcolumn, @field_array);
  my ($first, $last, $left, $right, $loadcmdconfig);
  my $i=0;
  my $j=0;

```

Η μεταβλητή `$keyword` παίρνει την τρέχουσα τιμή της λέξης κλειδί που διαβάστηκε και η `$field` παίρνει την τιμή του. Οι έγκυρες τιμές για τα πεδία ενός πίνακα της βάσης δεδομένων φυλάσσονται στην μεταβλητή `$validcolumn` και η μεταβλητή `@field_array` περιέχει τις τιμές των πεδίων σε συνάρτηση με τις στήλες του αρχείου. Οι τέσσερις μεταβλητές `$first`, `$last`, `$left` και `$right` είναι βοηθητικές στην περίπτωση που η σχέση των πεδίων του πίνακα της βάσης με τις στήλες του αρχείου χρειάζεται πιο περίπλοκη περιγραφή. Η μεταβλητή `$loadcmdconfig` κρατάει την εντολή που ανέγνωσε από το αρχείο και που θα εκτελέσει στην συνέχεια το πρόγραμμα με αυτά τα δεδομένα και τα `$i` και `$j` χρησιμοποιούνται για σε βρόχους επανάληψης.

```

  $loadconfig=$filename.".load" unless (defined $loadconfig);
  sysmsg("Reading $loadconfig load configuration file...\n",1);

```

Αν δεν έχει οριστεί από την γραμμή εντολών το όνομα του αρχείου, τότε αυτό θα είναι το ίδιο με το όνομα του αρχείου δεδομένων με την κατάληξη `.load`.

```

if (-e $loadconfig)
{
  if (-r $loadconfig)
  {
    open(LCFH,$loadconfig)
    or syserr("Couldn't read from $loadconfig:\nSystem error: $!",2);

    while (<LCFH>)
    {

```

Καθώς υπάρχει το αρχείο και έχει τις κατάλληλες άδειες για ανάγνωση, το ανοίγουμε για διάβασμα. Όσο υπάρχουν γραμμές στο αρχείο, εκτελούμε τα παρακάτω.

```

  if ($_ =~ /^#/ )
  {
    next;
  }

```

Οι γραμμές που ξεκινούν με '#' τις αγνοούμε ως σχόλια του αρχείου παραμετροποίησης.

```

  elsif ($_ =~ /\w/ )
  {

```

Αν η γραμμή έχει έστω και έναν έγκυρο χαρακτήρα, δηλαδή είναι μία μη κενή γραμμή, τότε εξετάζουμε τα παρακάτω.

```

  if ($_ =~ /$TARGETTABLE/i)
  {
    ($keyword, $tablename)=split(/=/, $_);
    chomp $tablename;
    $tablename=~tr/"'//d;
    $tablename=uc($tablename);
    $validcolumn=SelectData("tname", "tab", "tname", $tablename);
    if ($validcolumn!~/ $tablename/i)
    {
      syserr("Unknown table name \"$tablename\" in database:\n".
        "Load Config: $_",7);
    }
    $validcolumn=SelectData("column_name", "all_tab_columns",
      "table_name", $tablename);
  }
}

```

Αν η γραμμή που διαβάστηκε περιέχει την λέξη κλειδί TargetTable, τότε αναθέτουμε στις μεταβλητές \$keyword και \$tablename την λέξη κλειδί και την τιμή του αντίστοιχα. Αφαιρούμε από την τιμή που δίνει το όνομα του πίνακα της βάσης που θα φορτωθούν τα δεδομένα την αλλαγή γραμμής που περιέχει στο τέλος του, καθώς και τυχόν εισαγωγικά που μπορεί να έχει βάλει επιπλέον ο χρήστης και δεν χρειάζονται. Μετατρέπουμε το όνομα του πίνακα της βάσης σε κεφαλαία και στην μεταβλητή \$validcolumn αναθέτουμε το όνομα του πίνακα όπως το βρίσκουμε στη βάση. Αν αυτό υπάρχει με ένα select στο tname του view tab με βάση το όνομα του πίνακα που διαβάστηκε, θα μας επιστρέψει το όνομα του πίνακα πάλι, αν αυτό υπάρχει στη βάση ή κενό αν αυτό δεν υπάρχει. Συγκρίνουμε την τιμή που μας επιστράφηκε από την βάση με αυτή που έχει δοθεί ως όρισμα και αν αυτές είναι διαφορετικές, εκτυπώνεται στον χρήστη ένα μήνυμα λάθους. Η μεταβλητή \$validcolumn αρχικοποιείται σε μία λίστα χωριζόμενη με κόμμα, με τα πεδία που έχει αυτός ο πίνακας. Θα χρησιμοποιηθεί παρακάτω για να πιστοποιηθεί η ύπαρξη των πεδίων που θα δοθούν στο αρχείο παραμετροποίησης.

```

  elsif ($_ =~ /$ACTION/i)
  {
    ($keyword, $loadcmdconfig)=split(/=/, $_);

```

```

chomp $loadcmdconfig;
$loadcmdconfig=~tr/"'//d;
$loadcommand=$loadcmdconfig unless (defined $loadcommand);
}

```

Η ενέργεια που θα γίνει με τα δεδομένα του αρχείου που παραμετροποιείται, ελέγχεται σε αυτή την λογική διακλάδωση. Έτσι αν η πρόταση έχει την λέξη κλειδί για την ενέργεια, διαχωρίζουμε την λέξη αυτή από την τιμή της στις μεταβλητές \$keyword και \$loadcmdconfig. Στην συνέχεια αφαιρούμε την αλλαγή γραμμής που τυχόν να έχει στο τέλος, καθώς και τα μονά και διπλά εισαγωγικά. Θέτουμε την γενική μεταβλητή \$loadcmdconfig στη τιμή που διαβάστηκε από το αρχείο, μόνο όταν αυτή δεν έχει οριστεί προηγουμένως από την γραμμή εντολών.

```

elsif ($_~/ $DELIMITER/i)
{
($keyword, $delimiter)=split /=/, $_;
chomp $delimiter;
$delimiter=~tr/"'//d;
}

```

Όταν βρεθεί η λέξη κλειδί για τον διαχωριστή στηλών, αποθηκεύεται στην γενική μεταβλητή \$delimiter, αφού πρώτα γίνει η αφαίρεση χαρακτήρων που δεν χρειάζονται.

```

elsif ($_~/ $FIELDNUMBER/i)
{
($keyword, $fieldnumber)=split /=/, $_;
chomp $fieldnumber;
$fieldnumber=~tr/"'//d;
}

```

Ο αριθμός των στηλών που υπάρχουν στο αρχείο, καθορίζεται από την γενική μεταβλητή \$fieldnumber.

```

elsif ($_~/ $FUNCTION/i)
{
($keyword, $field)=split /=/, $_;
chomp $field;
$field=~s/$FUNCTION\[//i;
$field=~tr/\]//d;
$function{$keyword}=$field;
}

```

Μερικές στήλες, χρειάζονται μορφοποίηση για να εισαχθούν στη βάση δεδομένων, όπως για παράδειγμα η στήλη που δίνει την ημερομηνία. Η συνάρτηση που θα χρησιμοποιηθεί για την μορφοποίηση, δίνεται από την λέξη κλειδί 'Function' που μέσα σε αγκύλες που ακολουθούν έχει την ζητούμενη συνάρτηση μορφοποίησης. Η προσωρινή μεταβλητή \$field, παίρνει ότι ακολουθεί μετά την ισότητα του πεδίου και αφαιρεί επιπλέον την δεσμευμένη λέξη function και τις αγκύλες. Έτσι στο τέλος στην προσωρινή μεταβλητή έχει μείνει η συνάρτηση μορφοποίησης που κρατιέται στη γενική μεταβλητή %function με κλειδί το όνομα του πεδίου και τιμή την συνάρτηση.

```

elsif ($_~/ $FORMAT/i)
{
($keyword, $field)=split /=/, $_;
chomp $field;
$field=~s/$FORMAT\[//i;
$field=~tr/\]//d;
$format{$keyword}=$field;
}

```

Η συνάρτηση μορφοποίησης δέχεται και μία συγκεκριμένη διαμόρφωση (format) που πρέπει και αυτό να οριστεί και φυλάσσεται με τον ίδιο τρόπο στη γενική μεταβλητή %format.

```

elseif ($_~/ $FIELD/i or $_~/ $RANGE/i)
{
    ($keyword, $field)=split(/=/, $_);
    chomp $field;
    $field=~s/$FIELD\[//i;
    $field=~s/$RANGE\[//i;
    $field=~tr/\]//d;
    unless (defined $validcolumn)
    {
        syserr("$TARGETTABLE should be defined first in $filename.load",
            7);
    }
    if ($validcolumn!~/ $keyword/i)
    {
        syserr("Invalid column name $keyword for $tablename entity:\n".
            "Load Config: $_",7);
    }
}

```

Αν στην γραμμή που έχουμε διαβάσει υπάρχει η λέξη 'field' ή 'range', τότε αναθέτουμε στην προσωρινή μεταβλητή \$field τα πεδία που περιγράφονται μέσα στην αγκύλη, χωρίς την δεσμευμένη λέξη και τις αγκύλες. Για να εξετάσουμε αν το πεδίο του πίνακα της βάσης που δόθηκε υπάρχει στη βάση, χρησιμοποιούμε την μεταβλητή \$validcolumn που έχει οριστεί, με την δήλωση του πίνακα. Αν η δήλωση του πίνακα, δεν έχει προηγηθεί, τότε εκτυπώνεται ένα μήνυμα λάθους στον χρήστη. Μετά εξετάζεται και αν έχει δοθεί υπαρκτό όνομα για το πεδίο του πίνακα.

Πριν προχωρήσουμε στην ανάλυση των πεδίων της βάσης, θα πρέπει να περιγράψουμε την μορφή που μπορούν να έχουν αυτά. Για να γίνει εισαγωγή μίας εγγραφής στη βάση, θα πρέπει να έχουν οριστεί όλα τα πεδία που έχει ο πίνακας. Έτσι αν δεν έχει οριστεί έστω και ένα πεδίο, το πρόγραμμα προχωράει να διαβάσει και τα υπόλοιπα και κάνει εισαγωγή μόνο όταν έχουν οριστεί όλα. Τα πεδία παίρνουν ακέραιους αριθμούς που ξεκινούν από το 0 για την πρώτη στήλη. Αν ο αριθμός της στήλης είναι σκέτος ακέραιος αριθμός, τότε το πεδίο του πίνακα παίρνει την τιμή της στήλης που δηλώνει. Υπάρχουν περιπτώσεις που το αρχείο δεδομένων δεν δίνει πληροφορίες για όλα τα πεδία του πίνακα και θα πρέπει να δηλωθούν από εμάς. Αυτό γίνεται με τον τελεστή '->', δηλαδή η έκφραση 0->5, δηλώνει ότι για το πεδίο της θέσης 0, θα πάρει την τιμή 5. Με τον τελεστή '#' μπροστά από έναν ακέραιο, δηλώνει ότι το πεδίο δεν θα πάρει την τιμή που ακολουθεί, αλλά την τιμή της στήλης. Έτσι η έκφραση 2->#0, δηλώνει ότι το πεδίο για τη θέση 2, θα πάρει την τιμή που της στήλης 0 του αρχείου. Τέλος διακρίνονται περισσότερες από μία τιμές που μπορεί να πάρει ένα πεδίο από τις στήλες του αρχείου με κόμμα και διαδοχικά πεδία που παίρνουν την ίδια τιμή, μπορούν να δοθούν με δύο συνεχόμενες τελείες. Για παράδειγμα αν θέλουμε να περιγράψουμε ότι το πεδίο για τη θέση από 1 έως 5 θα πάρει την τιμή 9, θα γράφαμε 1->9, 2->9, 3->9, 4->9, 5->9, ενώ μπορούμε με την χρήση των δύο συνεχόμενων τελειών, να πούμε το ίδιο πράγμα και με 1..5->9.

```

@field_array=split(/,/, $field);
for ($i=0;$i<=#field_array;$i++)
{

```

Διακρίνουμε τα πεδία, αν υπάρχουν, με βάση το κόμμα στον πίνακα @field_array. Όσο έχουμε πεδία στον ίδιο πίνακα, εκτελούμε τα παρακάτω.

```

if ($field_array[$i]~/ ->/)
{
    ($left, $right)=split(/->/, $field_array[$i]);

```

Ελέγχουμε αν το πεδίο έχει τον τελεστή ανάθεσης '->' και καταχωρούμε τα δύο του μέρη στις μεταβλητές \$left και \$right.


```

if ($left=~/\.\./)
{
    ($first,$last)=split(/\.\./,$left);
    $first=int $first;
    $last=int $last;
    if ($first>$last)
    {
        syserr("The following configuration line, has greater ".
            "left than right limit:\n$_",7);
    }
    for ($j=$first;$j<=$last;$j++)
    {
        $table{$keyword}{$j}=QuoteStrings $right;
    }
}
else
{
    $table{$keyword}{$left}=QuoteStrings $right;
}
}
}

```

Ελέγχουμε το αριστερό πεδίο για την ύπαρξη του τελεστή των δύο τελειών. Αν υπάρχει τότε σπάμε το αριστερό πεδίο και αναθέτουμε τους δύο αριθμούς στις μεταβλητές \$first και \$last. Κάνουμε έναν έλεγχο για το αν έχει κατά λάθος δοθεί μεγαλύτερο ο πρώτος αριθμός και αν είναι εκτυπώνεται ένα μήνυμα λάθους και το πρόγραμμα τερματίζει. Αν είναι σωστά τα όρια, τότε με ένα απλό βρόχο for, αναθέτουμε στο διδιάστατο hash %table την ίδια τιμή για όλες τις θέσεις του πεδίου από \$first μέχρι \$last. Η γενική μεταβλητή %table έχει ως πρώτο κλειδί το όνομα του πεδίου του πίνακα, ως δεύτερο τον αριθμό της θέσης και ως τιμή την τιμή που έχει οριστεί στο αρχείο. Αν το αριστερό πεδίο δεν έχει τον τελεστή των δύο τελειών, τότε αναθέτουμε κατευθείαν την δεξιά τιμή στο %table.

```

else
{
    if ($field_array[$i] =~ /\.\./)
    {
        ($first,$last)=split(/\.\./,$left);
        $first=int $first;
        $last=int $last;
        if ($first>$last)
        {
            syserr("The following configuration line, has greater ".
                "left than right limit:\n$_",7);
        }
        for ($j=$first;$j<=$last;$j++)
        {
            $table{$keyword}{$j}="field";
        }
    }
    else
    {
        $table{$keyword}{$field_array[$i]}="field";
    }
}
}
}
}

```

Αν δεν υπάρχει ο τελεστής ανάθεσης '->', τότε ελέγχουμε μόνο την ύπαρξη του τελεστή '..' επαναλαμβάνοντας την διαδικασία που περιγράφηκε στην προηγούμενη παράγραφο. Η μόνη διαφορά είναι ότι σε αυτή την περίπτωση δεν αναθέτουμε εναλλακτική τιμή που έχει δώσει ο χρήστης, αλλά την λέξη "field" που δηλώνει ότι η τιμή είναι η τιμή της συγκεκριμένης στήλης του αρχείου.

```

        else
        {
            syserr("Unsupported configuration line:\n$_",7);
        }
    }
}
close(LCFH);

```

Στην περίπτωση που δεν βρούμε καμία από τις λέξεις κλειδιά που έχει το αρχείο παραμετροποίησης, εκτυπώνεται ένα μήνυμα λάθους και το πρόγραμμα τερματίζει, αλλιώς κλείνει το αρχείο και συνεχίζει.

```

foreach $keyword (keys %function)
{
    foreach $i (keys %{$table{$keyword}})
    {
        $table{$keyword}{$i}="function" if ($table{$keyword}{$i} eq "field");
    }
}

```

Αν υπάρχει μορφοποίηση με κάποια συνάρτηση, όπως είναι η `to_date` για παράδειγμα, τότε ελέγχουμε και την μεταβλητή `%function` που τις κρατάει για τα συγκεκριμένα πεδία και δίνουμε την τιμή "function", για να ξέρουμε όταν θα διαβάσουμε το αρχείο δεδομένων, ότι δεν θα αναθέσουμε κατευθείαν την τιμή, αλλά θα την διαμορφώσουμε με την συγκεκριμένη συνάρτηση που έχει οριστεί. Προφανώς γίνεται και έλεγχος ότι η μορφοποίηση έχει νόημα μόνο στα πεδία του πίνακα που παίρνουν την τιμή τους κατευθείαν από τις στήλες του αρχείου και όχι για τις θέσεις που έχουν πάρει σταθερές τιμές ή τιμές που εξαρτώνται από άλλες θέσεις και δηλώνονται με το '#'.

```

sysmsg("Load configuration file was read as following:\n",3);
sysmsg("$TARGETTABLE=$tablename\n",3);
sysmsg("$ACTION=$loadcmdconfig ($loadcommand)\n",3);
if (defined $loadcmdconfig);
sysmsg("$DELIMITER=$delimiter\n",3);
sysmsg("$FIELDNUMBER=$fieldnumber\n",3);
foreach $keyword (keys %function)
{
    sysmsg("function{$keyword}=$function{$keyword}\n",3);
    sysmsg("format{$keyword}=$format{$keyword}\n",3);
    if (exists $format{$keyword});
}
foreach $keyword (keys %table)
{
    foreach $field (keys %{$table{$keyword}})
    {
        sysmsg("table{$keyword}{$field}=$table{$keyword}{$field}\n",5);
    }
}
}
else
{
    syserr("Configuration file $loadconfig does not exist", 7);
}
}

```

Στο τέλος της ρουτίνας, εκτυπώνονται οι μεταβλητές όπως διαβάστηκαν από το αρχείο παραμετροποίησης του αρχείου δεδομένων και αν δεν υπάρχει το αρχείο, εκτυπώνεται ένα μήνυμα λάθους και τερματίζει.

4.2.2.10 Ρουτίνα επιλογής δεδομένων της βάσης SelectData

Η ρουτίνα SelectData εκτελεί ένα απλό SELECT στη βάση δεδομένων με βάση τα ορίσματα που της έχουν δοθεί.

```
# SelectData function. Error level 5.
```

```
sub SelectData
{
  my $columninfo;
  my $result;
  my @row;
```

Η μεταβλητή \$columninfo φτιάχνει το SELECT query που θα εκτελεστεί στη βάση, η \$result κρατάει το αποτέλεσμα που θα δώσει η εκτέλεση της ερώτησης και το @row κρατάει την γραμμή που επιστρέφει κάθε φορά η βάση.

```
if (defined $_[3])
{
  $columninfo="SELECT $_[0] FROM $_[1] WHERE $_[2]='$_[3]";
  sysmsg("Select query to execute in SelectData:\n",4);
  sysmsg($columninfo."\n",4);

  $sth=$dbh->prepare($columninfo) or
  syserr("Can't prepare SQL statement:\nDB error: $DBI::errstr", 5);

  $sth->execute or
  syserr("Can't execute SQL statement:\nDB error: $DBI::errstr", 5);
  $queries++;
```

Αν έχουμε τέσσερα ορίσματα περασμένα στην συνάρτηση, τότε δηλώνουμε την ερώτηση SELECT στη μεταβλητή \$columninfo και εκτυπώνουμε ένα μήνυμα αποσφαλμάτωσης με επίπεδο 4 για την ερώτηση που σχηματίστηκε. Προετοιμάζουμε την ερώτηση στο handler της βάσης δεδομένων που έχει ανοιχτεί στο κυρίως μέρος και βρίσκεται στη \$dbh μεταβλητή και το αναθέτουμε στη γενική μεταβλητή \$sth. Με την μέθοδο execute εκτελούμε την εντολή στη βάση. Σε περίπτωση λάθους κατά την προετοιμασία ή κατά την εκτέλεση, εκτυπώνεται ένα μήνυμα λάθους και το πρόγραμμα τερματίζει. Η μεταβλητή \$queries μετράει τον αριθμό των εντολών που εκτελούνται στη βάση για στατιστικούς λόγους.

```
while (@row=$sth->fetchrow_array)
{
  unless ($result)
  {
    $result=join " ", @row;
  }
  else
  {
    $result=$result.", $row[0]";
  }
}

sysmsg("Columns got of $tablename entity: $result\n",4);
}
```

Με την μέθοδο fetchrow_array επιστρέφεται κάθε φορά μία γραμμή αποτελεσμάτων από την βάση που αποθηκεύεται στην μεταβλητή @row. Καθώς υπάρχουν αποτελέσματα από την εκτέλεση της εντολής και αποθηκεύονται στη μεταβλητή \$result με τις τιμές να διαχωρίζονται με κόμμα και εκτυπώνεται με μήνυμα το αποτέλεσμα της εντολής.

```
else
```

```

    {
        syserror("Usage: SelectData(<column>, <table>, <column>, <data>)",5);
    }
    return $result;
}

```

Αν τα ορίσματα δεν είναι τέσσερα, τότε εκτυπώνεται ένα μήνυμα λάθους και το πρόγραμμα τερματίζει, αλλιώς επιστρέφεται το αποτέλεσμα που δημιουργήθηκε στην μεταβλητή \$result.

4.2.2.11 Ρουτίνα μετατροπής πολικών από καρτεσιανές συντεταγμένες AngleFromCartesian

Η ρουτίνα AngleFromCartesian υπολογίζει την γωνία σε πολικές συντεταγμένες από τις συντεταγμένες στους δύο άξονες που περνούν ως ορίσματα.

```
# AngleFromCartesian function. Error level 8.
```

```

sub AngleFromCartesian
{
    my ($phi,$x,$y,@arguments,$pi);

    @arguments=@_;

```

Η μεταβλητή \$phi κρατάει το αποτέλεσμα της γωνίας που θα υπολογιστεί. Τα \$x και \$y τις συντεταγμένες που περνάν ως παράμετροι. Το @arguments είναι ο πίνακας με τα ορίσματα και το \$pi είναι η σταθερά π. Ξεκινάει η ρουτίνα δεσμεύοντας τοπικά τα ορίσματα που έχουν περαστεί.

```

    if ($#arguments!=1)
    {
        syserr("Usage: <var>=AngleFromCartesian(<x>,<y>)",8);
    }

```

Αν δεν έχουμε δύο οπωσδήποτε ορίσματα, τότε εκτυπώνεται ένα σύντομο μήνυμα χρήσης και η ρουτίνα τερματίζει.

```

    else
    {
        $x=$arguments[0];
        $y=$arguments[1];
        $pi=atan2(1,1)*4;

        $phi=($y==0?90:atan2(abs($x),abs($y))/ $pi*180);

```

Εφόσον έχουμε δύο ορίσματα, τα αναθέτουμε στις τοπικές μεταβλητές που κρατούν τις καρτεσιανές συντεταγμένες. Η γωνία υπολογίζεται με βάση τι είναι το Y. Αν αυτό είναι μηδέν, τότε η γωνία είναι 90 μοίρες, διαφορετικά είναι $\text{toξοεφ}(|X|/|Y|)/\pi*180$.

```

    if ($y>=0 and $x<0)
    {
        $phi=360-$phi;
    }
    elsif ($y<0 and $x>=0)
    {
        $phi=180-$phi;
    }
    elsif ($y<0 and $x<0)
    {
        $phi=180+$phi;
    }

```

```
}
```

Ακολουθούν περιπτώσεις όπου η γωνία θα πρέπει να οριστεί με βάση το αν οι καρτεσιανές συντεταγμένες είναι θετικές ή αρνητικές.

```
$phi=sprintf("%.0f", $phi);
$phi=0 if ($phi==360);

return $phi;
}
```

Στρογγυλοποιούμε το αποτέλεσμα της πολικής γωνίας και το επιστρέφουμε στο τέλος της ρουτίνας.

4.2.2.12 Ρουτίνα εκτύπωσης στατιστικών PrintStatistics

Το πρόγραμμα διαθέτει μία ρουτίνα για να εκτυπώνει στατιστικά του χρόνου εκτέλεσης της ρουτίνας.

```
# PrintStatistics function. Error level 9.

sub PrintStatistics
{
    my (@arguments, $first, $last, $descr, $coltime, @timefmt);

    @arguments=@_;
```

Για τα ορίσματα χρησιμοποιείται η μεταβλητή @arguments. Οι μεταβλητές \$first και \$last είναι οι θέσεις του των γενικών μεταβλητών που κρατούν τις χρονικές υποδιαιρέσεις για την αρχική στιγμή και την τελική στιγμή μέτρησης. Η \$descr κρατάει την παράμετρο για μία σύντομη περιγραφή στην εκτύπωση των στατιστικών και οι \$coltime και @timefmt διαμορφώνουν την εκτύπωση του συνολικού χρόνου.

```
if ($#arguments!=2)
{
    syserr("Usage: PrintStatistics(<first>, <last>, <description>)", 9);
}
else
{
    $first=$arguments[0];
    $last=$arguments[1];
    $descr=$arguments[2];

    ($sec[$last], $min[$last], $hour[$last], $mday[$last])=gmtime;
```

Γίνεται έλεγχος για τον αριθμό των ορισμάτων και αν δεν βρεθεί ίσος με 3, εκτυπώνεται ένα μήνυμα λάθους και τερματίζει. Το \$# δίνει το τελευταίο στοιχείο ενός πίνακα που ξεκινά από την θέση 0. Διαφορετικά αρχικοποιούνται οι μεταβλητές της θέσης και της περιγραφής και καταγράφεται η τρέχουσα χρονική στιγμή.

```
if ($sec[$last]<$sec[$first])
{
    $sec[$last]=$sec[$last]+60;
    $sec[$last+1]=$sec[$last]-$sec[$first];
    $min[$last]--;
}
else
{
```

```

    $sec[$last+1]=$sec[$last]-$sec[$first];
}
if ($min[$last]<$min[$first])
{
    $min[$last]=$min[$last]+60;
    $min[$last+1]=$min[$last]-$min[$first];
    $hour[$last]--;
}
else
{
    $min[$last+1]=$min[$last]-$min[$first];
}
if ($hour[$last]<$hour[$first])
{
    $hour[$last]=$hour[$last]+24;
    $hour[$last+1]=$hour[$last]-$hour[$first];
    $mday[$last]--;
}
else
{
    $hour[$last+1]=$hour[$last]-$hour[$first];
}
}

```

Ακολουθεί η πράξη της αφαίρεσης των δύο χρονικών διαστημάτων. Του ενός που μετρήθηκε στη θέση \$last και του προηγούμενου που είχε κρατηθεί στη θέση \$first. Χρησιμοποιείται η θέση \$last+1 για να κρατήσει το αποτέλεσμα της διαφοράς των χρόνων. Λαμβάνεται υπόψη αν η τελευταία χρονική μονάδα είναι μικρότερη από την προηγούμενη, έτσι ώστε να δανειστεί μία χρονική μονάδα μεγαλύτερης τάξης για να γίνει η αφαίρεση και ταυτόχρονα αφαιρείται και η χρονική μονάδα που δανείστηκε.

```

$sec[$last+2]=$hour[$last+1]*3600+$min[$last+1]*60+$sec[$last+1];
$min[$last+2]=$hour[$last+1]*60+$min[$last+1]+$sec[$last+1]/60;
$hour[$last+2]=$hour[$last+1]+$min[$last+1]/60+$sec[$last+1]/3600;
$rate[$last]=(1000*$sec[$last+2])/$queries;
$rate[$last+1]=(1000*$min[$last+2])/$queries;
$rate[$last+2]=(1000*$hour[$last+2])/$queries;
if ($mday[$last]<$mday[$first])
{
    syserr("End month day is smaller, possible due to month change",0);
    syserr("Please ignore the following statistics",0);
}
}

```

Στη θέση \$last+2 κρατείται ο συνολικός χρόνος διάρκειας για την αντίστοιχη χρονική μονάδα. Στο @rate σε τρεις θέσεις από το \$last μέχρι το \$last+2, κρατείται ο ρυθμός εκτέλεσης ανά 1000 queries. Αν μεταξύ των δύο χρονικών στιγμών έχει αλλάξει η μέρα, τότε τα στατιστικά δεν είναι αξιόπιστα και εκτυπώνεται κατάλληλη προειδοποίηση στο χρήστη.

```

$coltime=sprintf("%d:%2.2d:%2.2d", $hour[$last+1], $min[$last+1],
$sec[$last+1]);
sysmsg("$descr statistics: $lines lines, $queries queries, ".
"$coltime total time\n",0);
$timefmt[0]=sprintf("%.4f", $rate[$last]);
$timefmt[1]=sprintf("%.4f", $rate[$last+1]);
$timefmt[2]=sprintf("%.4f", $rate[$last+2]);
sysmsg("Rate (1000 queries): $timefmt[0] secs, $timefmt[1] mins, ".
"$timefmt[2] hours\n",0);
$timefmt[3]=sprintf("%d", $sec[$last+2]);
$timefmt[4]=sprintf("%.4f", $min[$last+2]);
$timefmt[5]=sprintf("%.4f", $hour[$last+2]);
sysmsg("Total time: $timefmt[3] secs, $timefmt[4] mins, ".
"$timefmt[5] hours\n",0);
}

```

```
}
```

Στο τέλος διαμορφώνονται και στρογγυλοποιείται ο χρόνος εκτέλεσης και εκτυπώνεται στον χρήστη.

4.2.2.13 Ρουτίνα εισαγωγής δεδομένων InsertData

Για την εισαγωγή δεδομένων, δημιουργήθηκε η συνάρτηση InsertData και χρησιμοποιείται κυρίως από την συνάρτηση TimeSeriesConvert που θα περιγραφεί παρακάτω.

```
# InsertData function. Error level 10.

sub InsertData
{
  my $err=10;
  my (@arguments,$tseries,$tdate,$tval,$sth2,$query);

  @arguments=@_;
```

Η μεταβλητή \$err κρατάει το επίπεδο σφάλματος της ρουτίνας. Το @arguments κρατάει τις παραμέτρους της ρουτίνας, τα \$tseries, \$tdate, \$tval κρατούν την χρονοσειρά, την ημερομηνία και την τιμή αντίστοιχα. Το \$sth2 είναι ένας τοπικός χειριστής εντολών στη βάση και το \$query κρατάει την προς εκτέλεση εντολή εισαγωγής.

```
if ($#arguments==1)
{
  $tseries=$arguments[0];
  $tdate=$arguments[1];
  $query="INSERT INTO data_proc(series_id,data_date,status) VALUES".
  "($tseries,to_date($tdate,'YYYYMMDDHH24MISS'),'U')";
  sysmsg("Null query formed: $query\n",5);
  $sth2=$dbh->prepare($query) or
  syserr("Couldn't prepare value insertion in $tseries series:\n".
  "DB error: $DBI::errstr",$err);
  $sth2->execute or
  syserr("Couldn't execute value insertion in $tseries series:\n".
  "DB error: $DBI::errstr",$err);
  $queries++;
}
```

Αν έχουμε δύο ορίσματα, τότε είμαστε στην περίπτωση που η τιμή απουσιάζει και δεν εισάγουμε τίποτα στη θέση της.

```
elsif ($#arguments==2)
{
  $tseries=$arguments[0];
  $tdate=$arguments[1];
  $tval=$arguments[2];
  $query="INSERT INTO data_proc(series_id,data_date,status,value) VALUES".
  "($tseries,to_date($tdate,'YYYYMMDDHH24MISS'),'U',$tval)";
  sysmsg("Value query formed: $query\n",5);
  $sth2=$dbh->prepare($query) or
  syserr("Couldn't prepare value insertion in $tseries series:\n".
  "DB error: $DBI::errstr",$err);
  $sth2->execute or
  syserr("Couldn't execute value insertion in $tseries series:\n".
  "DB error: $DBI::errstr",$err);
  $queries++;
}
```

Αν έχουμε τρία ορίσματα, τότε έχουμε και τιμή και η εισαγωγή γίνεται ορίζοντας και το πεδίο value.

```

else
{
    syserr("Usage: InsertData(<series id>,<data date>[,<value>]"),$err);
}
}

```

Αν έχουμε οποιοδήποτε άλλο διαφορετικό αριθμό ορισμάτων, εκτυπώνεται ένα μήνυμα λάθους στον χρήστη και το πρόγραμμα τερματίζει.

4.2.2.14 Ρουτίνα μετατροπής χρονοσειρών TimeSeriesConvert

Η ρουτίνα TimeSeriesConvert ελέγχει τα πρωτογενή δεκάλεπτα εφαρμόζοντας έλεγχο ακραίων τιμών και χρονικής συνέπειας και μετατρέπει τις δεκάλεπτες χρονοσειρές σε ωριαίες, τις ωριαίες σε ημερήσιες, τις ημερήσιες σε μηνιαίες και τις μηνιαίες σε ετήσιες.

```
# TimeSeriesConvert function. Error level 11.
```

```

sub TimeSeriesConvert
{
    my $err=11;
    my (@arguments,$i,$j,$x,%timestep,$query,$ststh,$sth1,$ts,$prevt);
    my ($seriesout,$seriesin,%timext,$stepname,@varid);
    my ($variable,$sensor,@datarow,@result_array,@columns_array,@series);
    my ($mindate,$maxdate,$upperdate,$lowerdate,@datekeep,@sqlfunc,$value);
    my ($delta,$deltaback,@depend,$result,$columns,$nullify,%varname);
    my ($first,$last);
    my $STDDATEFORMAT="YYYYMMDDHH24MISS";
    my $COLDFORMAT="YYYY:MM:DD:HH24:MI:SS";

```

Η μεταβλητή \$err κρατάει το επίπεδο λάθους της συνάρτησης, ο πίνακας @arguments τα ορίσματα, τα \$i και \$j χρησιμοποιούνται ως βοηθητικές μεταβλητές σε βρόχους, το \$x χρησιμοποιείται προσωρινά για το προηγούμενο βήμα και το %timestep φυλάει τα χρονικά βήματα με κλειδί την ονομασία τους και τιμή το μοναδικό αναγνωριστικό αριθμό τους. Το \$query κρατάει την εντολή που σχηματίζεται κάθε φορά, τα \$ststh και \$sth1 χρησιμοποιούνται ως χειριστές των εντολών που εκτελούνται στη βάση, το \$ts είναι το χρονικό βήμα και το \$prevt είναι το προηγούμενο χρονικό βήμα.

Το αναγνωριστικό της χρονοσειράς εξόδου και εισόδου, δίνονται από τις μεταβλητές \$seriesout και \$seriesin. Το %timext έχει το τελικό χρονικό συμπλήρωμα του κάθε βήματος. Το \$stepname είναι το όνομα του χρονικού βήματος και ο πίνακας @varid κρατάει το όνομα του χρονικού βήματος. Η μεταβλητή \$variable κρατάει τον αναγνωριστικό αριθμό της μετεωρολογικής μεταβλητής, ο \$sensor του αισθητήρα και τα @datarow, @result_array και @columns_array είναι βοηθητικές μεταβλητές για τα αποτελέσματα που επιστρέφουν οι εκτέλεση εντολών στη βάση. Ο πίνακας @series κρατάει τα αναγνωριστικά όλων των χρονοσειρών που θα ελεγχθούν ή θα μετατραπούν. Τα \$mindate και \$maxdate αρχικοποιούνται στην ελάχιστη και την μέγιστη ημερομηνία αντίστοιχα της κάθε χρονοσειράς που εξετάζεται. Τα \$lowerdate και \$upperdate είναι το χρονικό παράθυρο που συλλέγει τα δεδομένα για μετατροπή στο προκαθορισμένο χρονικό βήμα. Το @datekeep είναι προσωρινή μεταβλητή για τις ημερομηνίες, το @sqlfunc ορίζει την συνάρτηση που θα πρέπει να υλοποιηθεί για την μετατροπή και το \$value είναι η τιμή της μετατροπής.

Τα \$delta και \$deltaback είναι οι χρονική ολίσθηση εμπρός και πίσω, σύμφωνα με το χρονικό βήμα που χρησιμοποιείται κάθε φορά. Ο πίνακας @depend φανερώνει εξαρτήσεις στην μετατροπή χρονοσειρών από άλλες τρίτες. Το \$result και το \$column είναι βοηθητικές μεταβλητές για την μετατροπή, το \$nullify είναι μία λογική μεταβλητή για τον μηδενισμό μίας μη συμπληρωμένης τιμής και το %varname κρατάει όλες τις ακραίες τιμές και τις τιμές απόκλισης για τον έλεγχο συνέπειας.

Τα \$first και \$last χρησιμοποιούνται για ανάλυση του τελεστή '..' για συνεχόμενες τιμές χρονοσειρών στη γραμμή εντολών. Οι ακόλουθες δύο μεταβλητές, χρησιμοποιούν δύο διαφορετικές μορφοποιήσεις της ημερομηνίας.

```
my $MIN_TEMP=-30;
my $MAX_TEMP=60;
my $TEMP_DIFF=20;
my $MIN_HUM=0;
my $MAX_HUM=100;
my $HUM_DIFF=30;
my $MIN_WIND=0;
my $MAX_WIND=60;
my $WIND_DIFF=30;
my $MIN_WIND_DIRECTION=0;
my $MAX_WIND_DIRECTION=360;
my $WIND_DIRECTION_DIFF=360;
my $MIN_PRESSURE=910-20;
my $MAX_PRESSURE=1080-20;
my $MIN_SEA_PRESSURE=910;
my $MAX_SEA_PRESSURE=1080;
my $PRESSURE_DIFF=25;
my $MIN_RAIN=0;
my $MAX_RAIN=50;
my $RAIN_DIFF=50;
my $MIN_SOLAR=-1;
my $MAX_SOLAR=1500;
my $SOLAR_DIFF=1501;
my $MIN_TOTAL_SOLAR=0;
my $MAX_TOTAL_SOLAR=3000;
my $TOTAL_SOLAR_DIFF=3000;
my $MIN_SUNSHINE=0;
my $MAX_SUNSHINE=10;
my $SUNSHINE_DIFF=10;
```

Οι παραπάνω σταθερές είναι τα όρια για τον έλεγχο συνέπειας για κάθε μετεωρολογική μεταβλητή που εξετάζεται. Αυτά έχουν παρθεί από τα στανταρ του World Meteorological Organisation (WMO), αλλά πολλές τιμές έχουν πειραχθεί για να προσεγγίζουν τις πραγματικές ανάγκες του σταθμού.

```
@arguments=@_;

syserr("Usage: TimeSeriesConvert(<step>)", $err) unless ($#arguments==0);

$stepname=$arguments[0];
$stepname=~s/ly$/i;
$stepname="Day" if ($stepname eq "Dai");
```

Ξεκινάει το πρόγραμμα αντιγράφοντας όλα τα ορίσματα σε έναν πίνακα @arguments. Ελέγχεται αν έχουν δοθεί ο σωστός αριθμός ορισμάτων και αν όχι, εκτυπώνεται ένα μήνυμα λάθους. Το χρονικό βήμα βγαίνει από την περιγραφή της χρονοσειράς, αφαιρώντας την κατάληξη ly και μετατρέποντας το Dai σε Day για την περίπτωση της ημερήσιας.

```
$query="SELECT step_id,name FROM time_step";
RunQuery($query,"select on time step",$err);
while (@datarow=$sth->fetchrow_array)
{
    $timestep{$datarow[1]}=$datarow[0];
}
syserr("While fetching time step data:\nDB error: $DBI::errstr",$err)
if ($DBI::err);
```

Αρχικοποιείται η μεταβλητή %timestep με τα χρονικά βήματα που υπάρχουν στη βάση.

```

$query="SELECT variable_id,name FROM variable";
RunQuery($query,"variable selection query",$err);
while (@datarow=$sth->fetchrow_array)
{
  if ($datarow[1]=~/temperature/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_TEMP;
    $varname{$datarow[0]}{'max'}=$MAX_TEMP;
    $varname{$datarow[0]}{'diff'}=$TEMP_DIFF;
  }
  elsif ($datarow[1]=~/humidity/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_HUM;
    $varname{$datarow[0]}{'max'}=$MAX_HUM;
    $varname{$datarow[0]}{'diff'}=$HUM_DIFF;
  }
  elsif ($datarow[1]=~/wind speed/i or $datarow[1]=~/wind gust/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_WIND;
    $varname{$datarow[0]}{'max'}=$MAX_WIND;
    $varname{$datarow[0]}{'diff'}=$WIND_DIFF;
  }
  elsif ($datarow[1]=~/wind direction/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_WIND_DIRECTION;
    $varname{$datarow[0]}{'max'}=$MAX_WIND_DIRECTION;
    $varname{$datarow[0]}{'diff'}=$WIND_DIRECTION_DIFF;
  }
  elsif ($datarow[1]=~/pressure/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_PRESSURE;
    $varname{$datarow[0]}{'max'}=$MAX_PRESSURE;
    $varname{$datarow[0]}{'diff'}=$PRESSURE_DIFF;
  }
  elsif ($datarow[1]=~/rainfall/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_RAIN;
    $varname{$datarow[0]}{'max'}=$MAX_RAIN;
    $varname{$datarow[0]}{'diff'}=$RAIN_DIFF;
  }
  elsif ($datarow[1]=~/total solar/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_TOTAL_SOLAR;
    $varname{$datarow[0]}{'max'}=$MAX_TOTAL_SOLAR;
    $varname{$datarow[0]}{'diff'}=$TOTAL_SOLAR_DIFF;
  }
  elsif ($datarow[1]=~/solar/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_SOLAR;
    $varname{$datarow[0]}{'max'}=$MAX_SOLAR;
    $varname{$datarow[0]}{'diff'}=$SOLAR_DIFF;
  }
  elsif ($datarow[1]=~/sunshine/i)
  {
    $varname{$datarow[0]}{'min'}=$MIN_SUNSHINE;
    $varname{$datarow[0]}{'max'}=$MAX_SUNSHINE;
    $varname{$datarow[0]}{'diff'}=$SUNSHINE_DIFF;
  }
  $varid[$datarow[0]]=$datarow[1];
}
syserr("While fetching variable data:\nDB error: $DBI::errstr",$err)
if ($DBI::err);

```

Αρχικοποιείται η μεταβλητή @varid στα ονόματα και τα αναγνωριστικά των μεταβλητών, καθώς και η μεταβλητή %varname στα μετεωρολογικά όρια.

```
if ($timeseries ne "all")
{
    $result=$timeseries;
    @result_array=split /,/, $result;
    for $i (0..$#result_array)
    {
        if ($result_array[$i]=~/\.\./)
        {
            ($first,$last)=split /\.\./, $result_array[$i];
            syserr("Left time series limit $first is greater than $last", $err)
            if ($first>$last);
            for $j ($first..$last)
            {
                push @series, $j;
            }
        }
        else
        {
            push @series, $result_array[$i];
        }
    }
}
```

Αν δεν έχει δοθεί η λέξη "all" που σημαίνει να εκτελέσει έλεγχο ή μετατροπή σε όλες τις χρονοσειρές, εξετάζεται η γενική μεταβλητή \$timeseries για να πάρουμε την ακριβή λίστα των αναγνωριστικών των χρονοσειρών στον πίνακα @series.

```
if ($arguments[0]=~/normalised/i)
{
    sysmsg("Making normalised ten-minute data...\n",0);

    $query="SELECT series_id,variable_id,sensor_id FROM time_series ".
    "WHERE type_id=1 AND step_id=1";
    if (defined $series[0])
    {
        $query=$query." AND series_id=$series[0]";
        for $i (1..$#series)
        {
            $query=$query." OR series_id=$series[$i]";
        }
    }
    sysmsg("Info query: $query\n",5);
    $sth1=$dbh->prepare($query) or
    syserr("Can't prepare normilised info query:\nDB error: $DBI::errstr", $err);
    $sth1->execute or
    syserr("Can't execute normilised info query:\nDB error: $DBI::errstr", $err);
    $queries++;
}
```

Γίνεται έλεγχος αν έχει δοθεί παράμετρος για κανονικοποίηση ως προς την συνέπεια των τιμών. Ετοιμάζεται μία εντολή SQL για να πάρει όλες τις πρωτογενείς χρονοσειρές. Αν έχει οριστεί το @series, τότε χρησιμοποιείται και αυτό για να επιλεγθούν οι απαιτούμενες πληροφορίες μόνο για αυτές.

```
while (@datarow=$sth1->fetchrow_array)
{
    $seriesin=$datarow[0];
    $variable=$datarow[1];
    $sensor=$datarow[2];
}
```

Καθώς υπάρχουν χρονοσειρές, αρχικοποιούμε την χρονοσειρά εισόδου, την μεταβλητή και τον αισθητήρα.

```
$query="SELECT series_id FROM time_series WHERE variable_id=$variable ".
"AND sensor_id=$sensor AND type_id=2 AND step_id=1";
RunQuery($query,"series ID statement",$err);
@result_array=$sth->fetchrow_array;
syserr("Unable to fetch series ID",$err) if ($DBI::err);
$seriesout=$result_array[0];
syserr("No $arguments[0] time series found in database",$err)
unless (defined $seriesout);
```

Αρχικοποιούμε και το \$seriesout, για να δώσει την χρονοσειρά εξόδου.

```
$query="SELECT to_char(min(data_date),'$STDDATEFORMAT') ".
"FROM data_raw WHERE series_id=$seriesin";
RunQuery($query,"min data statement",$err);
@result_array=$sth->fetchrow_array;
syserr("Unable to fetch min date",$err) if ($DBI::err);
$datekeep[0]=$result_array[0];
syserr("No minimum date returned",$err) unless (defined $datekeep[0]);

$query="SELECT to_char(max(data_date),'$STDDATEFORMAT') FROM data_raw ".
"WHERE series_id=$seriesin";
RunQuery($query,"max date statement",$err);
@result_array=$sth->fetchrow_array;
syserr("Unable to fetch max date",$err) if ($DBI::err);
$datekeep[1]=$result_array[0];
syserr("No maximum date returned",$err) unless (defined $datekeep[1]);

$mindate=$datekeep[0];
$maxdate=$datekeep[1];

sysmsg("Mindate=\"$mindate\" and maxdate=\"$maxdate\"\n",4);
```

Η ελάχιστη και η μέγιστη ημερομηνία που υπάρχουν δεδομένα για την συγκεκριμένη χρονοσειρά ανακτάται από τις δύο παραπάνω εντολές SQL που εκτελούνται στη βάση. Αποθηκεύονται στην προσωρινή μεταβλητή @datekeep και στην συνέχεια ανατίθενται στις \$mindate και \$maxdate και εκτυπώνονται στην οθόνη.

```
$query="SELECT to_char(data_date,'$STDDATEFORMAT'),value FROM data_raw ".
"WHERE series_id=$seriesin ".
"AND data_date>=to_date($mindate,'$STDDATEFORMAT') ".
"AND data_date<=to_date($maxdate,'$STDDATEFORMAT') ".
"ORDER BY data_date";
sysmsg("Get all query: $query\n",5);
RunQuery($query,"value statement in $seriesin time series",$err);

sysmsg("Producing $seriesout $varid[$variable] time series: limits ".
"$varname{$variable}{'min'}..$varname{$variable}{'max'} and difference ".
"$varname{$variable}{'diff'}\n",0);
```

Ερώτηση ανάκτησης όλων των εγγραφών που βρίσκονται στο χρονικό διάστημα της ελάχιστης και της μέγιστης ημερομηνίας.

```
while (@columns_array=$sth->fetchrow_array)
{
  if (defined $columns_array[1])
  {
    $result=$columns_array[1];
    $columns=$result unless (defined $columns);
```

Καθώς υπάρχουν δεδομένα, συνέχισε την μετατροπή. Αν είναι ορισμένη η τιμή της χρονοσειράς εισόδου, τότε ανάθεσέ την στη `$result` καθώς και την `$columns` που κρατάει την προηγούμενη τιμή και παίρνει την ίδια τιμή με την τρέχουσα μόνο στην πρώτη εγγραφή που δεν έχει οριστεί.

```
if ($result<$varname{$variable}{'min'} or
    $result>$varname{$variable}{'max'})
{
    syserr("Check violation in $seriesout $varid[$variable] ".
        "time series:\nRejected ".
        "[${columns_array[0]}, $result]: Out of marginal ".
        "limits $varname{$variable}{'min'}..$varname{$variable}{'max'}",0);
    InsertData($seriesout,$columns_array[0]);
}
```

Αν η τιμή είναι εκτός των ακραίων τιμών, τότε εκτύπωσε μία προειδοποίηση για λάθος και συνέχισε προσθέτοντας εγγραφή αγνοώντας την τιμή.

```
else
{
    if (abs($result-$columns)>$varname{$variable}{'diff'})
    {
        syserr("Check violation in $seriesout $varid[$variable] ".
            "time series:\nRejected ".
            "[${columns_array[0]}, $result]: Out of marginal ".
            "difference |$result-$columns|>$varname{$variable}{'diff'}",0);
        InsertData($seriesout,$columns_array[0]);
    }
}
```

Διαφορετικά αν η διαφορά με την προηγούμενη συνεχόμενη τιμή είναι μεγαλύτερη από την αναμενόμενη, τότε απορρίπτεται η τιμή και δεν ορίζεται κατά την εισαγωγή.

```
else
{
    InsertData($seriesout,$columns_array[0],$result);
    $columns=$result;
}
}
```

Διαφορετικά αν περάσει και τους δύο ελέγχους, είναι έγκυρη και εισάγεται κανονικά στη βάση.

```
else
{
    InsertData($seriesout,$columns_array[0]);
}
$lines++;
}
undef $columns;
sysmsg("Finished $seriesout $arguments[0] time series from $seriesin".
    "...\n",0);
}
```

Αλλιώς αν δεν έχει οριστεί τιμή, γίνεται εισαγωγή εγγραφής χωρίς την τιμή αυτή. Το `$lines` αυξάνεται κατά ένα, το `$columns` γίνεται αόριστο και εκτυπώνεται μήνυμα ενημερώνοντας τον χρήστη ότι ο έλεγχος της χρονοσειράς ολοκληρώθηκε.

```
else
{
    sysmsg("Making normalised $timestep data...\n",0);
    $ts=$timestep{$arguments[0]};
```

```

$prevts=$ts-1;
$timext{"Hourly"}="0000";
$timext{"Daily"}="080000";
$timext{"Monthly"}="01000000";
$timext{"Yearly"}="1001000000";

```

Αλλιώς αν δεν είναι για έλεγχο, θα είναι για μετατροπή χρονοσειράς από το ένα χρονικό βήμα στο άλλο. Το χρονικό βήμα ορίζεται στο ακέραιο αναγνωριστικό του από το %timestep και το προηγούμενο χρονικό βήμα είναι κατά ένα μικρότερο. Το επίθεμα του χρόνου ορίζεται εδώ στη μεταβλητή %timext.

```

$x=$prevts;
for $i
($x."04"..$x."07", $x."09"..$x."11", $x."13", $x."15"..$x."16", $x."18", $x."20")
{ $sqlfunc[$i]="AVG"; }
$sqlfunc[$x."01"]="SUM";
$sqlfunc[$x."02"]="SUM";
$sqlfunc[$x."03"]="ARC";
$sqlfunc[$x."08"]="SUM";
$sqlfunc[$x."12"]="SUM";
$sqlfunc[$x."14"]="SUM";
$sqlfunc[$x."17"]="ARC";
$sqlfunc[$x."19"]="MAX";
$sqlfunc[$x."21"]="MAX";
$delta=ParseDateDelta("+ 1 $stepname");
$deltaback=ParseDateDelta("- 1 $stepname");
$depend[$x."03"]=$x."04";
$depend[$x."17"]=$x."18";

```

Σε αυτό το σημείο, αρχικοποιείται ο πίνακας @sqlfunc σε AVG, SUM, ARC, MAX ανάλογα αν η μεταβλητή είναι για μέση τιμή, αθροιστική, για γωνία σε μοίρες ή μέγιστα αντίστοιχα. Το \$delta παίρνει ένα βήμα παραπάνω και το \$deltaback παίρνει μείων ένα χρονικό βήμα. Έχουμε και δύο εξαρτήσεις για τις χρονοσειρές της διεύθυνσης ανέμου, που πρέπει να έχουμε και την ταχύτητα για τον υπολογισμό του διανύσματος.

```

$query="SELECT series_id,variable_id,sensor_id FROM time_series ".
"WHERE type_id=2 AND step_id=$prevts";
if (defined $series[0])
{
  $query=$query." AND series_id=$series[0]";
  for $i (1..$#series)
  {
    $query=$query." OR series_id=$series[$i]";
  }
}
$query=$query." ORDER BY series_id";
sysmsg("Info query: $query\n",5);
$tsth=$dbh->prepare($query) or
syserr("Unable to prepare $arguments[0] time series select:\nDB error: ".
"$DBI::errstr",11);
$tsth->execute or
syserr("Unable to execute $arguments[0] time series select:\nDB error: ".
"$DBI::errstr",11);
$queryes++;

```

Εκτελείται εντολή SQL για να πάρουμε πληροφορίες για την προηγούμενη χρονοσειρά, όπως είναι το αναγνωριστικό της, η μεταβλητή που καταγράφει και ο αισθητήρας. Ελέγχεται και αν έχει οριστεί το @series, που σημαίνει ότι θα πρέπει να ανακτηθούν πληροφορίες για τις συγκεκριμένες χρονοσειρές που έχει δηλώσει ο χρήστης.

```

while (@datarow=$tsth->fetchrow_array)
{

```

```

$seriesin=$datarow[0];
$variable=$datarow[1];
$sensor=$datarow[2];

```

Καθώς υπάρχουν εγγραφές με χρονοσειρές, αρχικοποιούνται η χρονοσειρά εισόδου, η μεταβλητή και ο αισθητήρας,

```

$query="SELECT series_id FROM time_series WHERE variable_id=$variable ".
"AND sensor_id=$sensor AND step_id=$ts";
$sth1=$dbh->prepare($query)
or syserr("Unable to prepare series ID statement:\nDB error: ".
"$DBI::errstr",11);
$sth1->execute
or syserr("Unable to execute series ID statmenent:\nDB error: ".
"$DBI::errstr",11);
$queryes++;
@result_array=$sth1->fetchrow_array;
syserr("Unable to fetch series ID",11) if ($DBI::err);
$seriesout=$result_array[0];
syserr("No $arguments[0] time series found in database",11)
unless (defined $seriesout);

```

Βρίσκουμε στη βάση δεδομένων την αντίστοιχη χρονοσειρά εξόδου.

```

$query="SELECT to_char(min(data_date),'$COLDATEFORMAT') ".
"FROM data_proc WHERE series_id=$seriesin";
$sth1=$dbh->prepare($query)
or syserr("Unable to prepare min date statement:\nDB error: ".
"$DBI::errstr",11);
$sth1->execute
or syserr("Unable to execute min date statmenent:\nDB error: ".
"$DBI::errstr",11);
$queryes++;
@result_array=$sth1->fetchrow_array;
syserr("Unable to fetch min date",11) if ($DBI::err);
$datekeep[0]=$result_array[0];
syserr("No minimum date returned",11) unless (defined $datekeep[0]);

$query="SELECT to_char(max(data_date),'$STDDATEFORMAT') FROM data_proc ".
"WHERE series_id=$seriesin";
$sth1=$dbh->prepare($query)
or syserr("Unable to prepare max date statement:\nDB error: ".
"$DBI::errstr",11);
$sth1->execute
or syserr("Unable to execute max date statmenent:\nDB error: ".
"$DBI::errstr",11);
$queryes++;
@result_array=$sth1->fetchrow_array;
syserr("Unable to fetch max date",11) if ($DBI::err);
$datekeep[1]=$result_array[0];
syserr("No maximum date returned",11) unless (defined $datekeep[1]);

```

Παίρνουμε την ελάχιστη και την μέγιστη ημερομηνία που υπάρχουν εγγραφές για την χρονοσειρά εισόδου. Το ελάχιστο έχει επιστραφεί με ':' για να γίνει εύκολα ο διαχωρισμός παρακάτω.

```

@result_array=split /:/$datekeep[0];
$maxdate=$datekeep[1];
$mindate=$result_array[0];
for ($i=1;$i<=5-$ts;$i++)
{
    $mindate=$mindate.$result_array[$i];
}

```

```

$lowerdate=$mindate;
for($i=$i;$i<=5;$i++)
{
    $mindate=$mindate.$result_array[$i];
}
$lowerdate=$lowerdate.$timext{$arguments[0]};
if ($mindate<=$lowerdate)
{
    $upperdate=$lowerdate;
    $lowerdate=DateCalc($lowerdate,$deltaback);
    $lowerdate=~tr://d;
}
else
{
    $upperdate=DateCalc($lowerdate,$delta);
    $upperdate=~tr://d;
}
sysmsg("Converting $seriesout $varid[$variable] time series ".
"($mindate-$maxdate)\n", 0);
sysmsg("First lower=\"$lowerdate\" and upper=\"$upperdate\" dates\n",4);

```

Διαχωρίζεται η ελάχιστη ημερομηνία στις χρονικές της υποδιαιρέσεις, ενώ η μέγιστη ανατίθεται κατευθείαν στην \$maxdate. Η ελάχιστη ημερομηνία συντίθεται μέχρι το σημείο της ακρίβειας του συγκεκριμένου χρονικού βήματος. Το μέρος αυτό ανατίθεται στο κάτω όριο του χρονικού παραθύρου. Η ελάχιστη ημερομηνία παίρνει την πλήρη τιμή της, ενώ το κάτω όριο παίρνει την προέκταση του χρόνου για το συγκεκριμένο βήμα. Αν η ελάχιστη θερμοκρασία είναι μικρότερη ή ίση από το κάτω όριο, τότε το πάνω όριο γίνεται ίσο με το κάτω και το ίδιο κατεβαίνει ένα χρονικό βήμα, αλλιώς το πάνω όριο αυξάνει ένα χρονικό βήμα σε σχέση με το κάτω. Εκτυπώνεται και ένα κατατοπιστικό μήνυμα για τους χρόνους πως διαμορφώθηκαν.

```

if ($sqlfunc[$seriesin] eq "ARC")
{
    $query="SELECT to_char(f.data_date,'$STDDATEFORMAT'),f.value,v.value ".
"FROM data_proc f,data_proc v WHERE f.series_id=$seriesin ".
"AND v.series_id=$depend[$seriesin] AND f.data_date=v.data_date ".
"AND f.data_date>=to_date($mindate,'$STDDATEFORMAT') ".
"AND f.data_date<=to_date($maxdate,'$STDDATEFORMAT') ".
"ORDER BY f.data_date";
}
else
{
    $query="SELECT to_char(data_date,'$STDDATEFORMAT'),value ".
"FROM data_proc WHERE series_id=$seriesin ".
"AND data_date>=to_date($mindate,'$STDDATEFORMAT') ".
"AND data_date<=to_date($maxdate,'$STDDATEFORMAT') ".
"ORDER BY data_date";
}
sysmsg("Get all query: $query\n",5);
$sth1=$dbh->prepare($query) or
syserr("Unable to prepare value statement in $seriesin series",11);
$sth1->execute or
syserr("Unable to execute value statement in $seriesin series",11);
$queryes++;
$result=0;
$columns=0;
>nullify="no";

```

Η ερώτηση για να παρθούν όλες οι τιμές της χρονοσειράς εισόδου, διαφέρει στην περίπτωση που είναι για την διεύθυνση ανέμου, γιατί τότε θα πρέπει να πάρουμε υπόψη και την ταχύτητα.

```

while (@columns_array=$sth1->fetchrow_array)
{

```



```

sysmsg("Dates: $lowerdate<$columns_array[0]<=$upperdate, ".
"Value: ".DBI::neat($columns_array[1])."\n",5);
if ($columns_array[0]<=$lowerdate or $columns_array[0]>$upperdate)
{

```

Καθώς έχουμε εγγραφές, έλεγξε αν η ημερομηνία της εγγραφής είναι έξω από το χρονικό παράθυρο που εξετάζουμε.

```

if ($nullify eq "yes")
{
  InsertData($seriesout,$upperdate);
}

```

Αν είναι εκτός και η λογική σημαία για μηδενισμό της τιμής έχει γίνει αλήθεια, η εγγραφή εισάγεται χωρίς την τιμή της.

```

else
{
  if ($sqlfunc[$seriesin] eq "AVG")
  {
    $value=$result/$columns;
  }
  elseif ($sqlfunc[$seriesin] eq "SUM" or
    $sqlfunc[$seriesin] eq "MAX")
  {
    $value=$result;
  }
  elseif ($sqlfunc[$seriesin] eq "ARC")
  {
    $value=AngleFromCartesian($result_array[0],$result_array[1]);
  }
  InsertData($seriesout,$upperdate,$value);
}
}

```

Αλλιώς αν δεν πρόκειται να αγνοηθεί, υπολογίζεται η μέση τιμή, το άθροισμα, το μέγιστο, η γωνία και εισάγονται στη βάση.

```

if ($columns_array[0]>=$upperdate)
{
  $lowerdate=$upperdate;
  $upperdate=DateCalc($upperdate,$delta);
  $upperdate=~tr://d;
  while ($columns_array[0]>$upperdate)
  {
    InsertData($seriesout,$upperdate);
    $lowerdate=$upperdate;
    $upperdate=DateCalc($upperdate,$delta);
    $upperdate=~tr://d;
  }
}
$result=0;
$columns=0;
@result_array=();
>nullify="no";
}

```

Αν η ημερομηνία της τρέχουσας εγγραφής είναι μεγαλύτερη ή ίση με το πάνω όριο, τότε το αυξάνουμε συνέχεια κατά ένα χρονικό βήμα, μέχρι να το ξεπεράσει και αρχικοποιούμε τις μεταβλητές.

```

    if (defined $columns_array[1])
    {
        if ($sqlfunc[$seriesin] eq "AVG" or $sqlfunc[$seriesin] eq "SUM")
        {
            $result+=$columns_array[1];
        }
        elsif ($sqlfunc[$seriesin] eq "MAX")
        {
            $result=$columns_array[1] if ($columns_array[1]>$result);
        }
        elsif ($sqlfunc[$seriesin] eq "ARC")
        {
            if (defined $columns_array[2])
            {
                $result_array[0]+=$columns_array[2]*sin($columns_array[1]);
                $result_array[1]+=$columns_array[2]*cos($columns_array[1]);
            }
            else
            {
                $nullify="yes";
            }
        }
        $columns++;
    }
    else
    {
        $nullify="yes";
    }
    $lines++;
}
sysmsg("Finished $seriesout $arguments[0] time series from $seriesin".
"... \n",0);
}
}
}

```

Αν έχει οριστεί η τιμή, τότε με βάση την περίπτωση που βρισκόμαστε, υπολογίζουμε το άθροισμα, το μέγιστο και το άθροισμα των συντεταγμένων. Σε περίπτωση που δεν έχει οριστεί η τιμή, η λογική μεταβλητή \$nullify παίρνει την τιμή "yes", έτσι ώστε να μηδενιστεί και ολόκληρη η εγγραφή της μετατροπής.

4.2.2.15 Ρουτίνα παραγωγής γραφημάτων GenerateGraphs

Η ρουτίνα αυτή χρησιμοποιείται για την παραγωγή των διαγραμμάτων των μετεωρολογικών μεταβλητών.

```
# GenerateGraphs function. Error level 12.
```

```

sub GenerateGraphs
{
    my $err=12;
    my (%stepid,$query,@row,@datarow,@arguments,$ts,@series,@result_array);
    my ($result,$first,$last,$tsth,$i,$j,$delta,%timelength,$fontheight,$trans);
    my (@columns_array,$columns,$image,%color,$imagefile,%varname,%stepname);
    my ($language,$maxdate,$mindate,$unit,$styled,$start,$stop,$step,$seriesstep);
    my ($date,$x,$y,$timelabel,$font,$fontwidth,%timestep,$deltastep,%zeromark);
    my ($low,$high,$minval,$maxval,$avgval,$varvalue,$vardate);
    my ($xp,$yp,%timeformat,$minvaldate,$maxvaldate);
}

```

Το `$err` είναι το επίπεδο λάθους. Το `%stepid` κρατάει το πλήρες όνομα του χρονικού βήματος, το `$query` την προς εκτέλεση εντολή SQL, τα `@row` και `@datarow` τις επιστρεφόμενες γραμμές αποτελεσμάτων, το `@arguments` τα ορίσματα, το `$ts` το χρονικό βήμα, το `@series` τις χρονοσειρές και το `@result_array` βοηθητικό για ενδιάμεσα αποτελέσματα.

Το `$result` και αυτό βοηθητικό, το `$first` και `$last` χρησιμοποιούνται για την ανάλυση του ``.`,` το `$tsth` είναι χειριστής των εντολών SQL, τα `$i` και `$j` χρησιμοποιούνται στους βρόχους και το `$delta` είναι η χρονική αύξηση. Το `%timelength` χρησιμοποιείται για να ορίσει το χρονικό παράθυρο, το `$fontheigh` για το ύψος των γραμματοσειρών και το `$trans` για το διάφανο στυλ των γραμμών. Το `@columns_array` είναι βοηθητική μεταβλητή, όπως και το `$columns`, το `$image` είναι το αντικείμενο του διαγράμματος, το `%color` είναι για τα χρώματα, το `$imagefile` είναι το τελικό αποτέλεσμα του διαγράμματος σε αρχείο, το `%varname` και `%stepname` είναι τα ονόματα των μεταβλητών και του χρονικού βήματος αντίστοιχα.

Το `$language` είναι η γλώσσα που θα χρησιμοποιηθεί, το `$maxdate` και `$mindate` είναι η μέγιστη και ελάχιστη ημερομηνία μίας χρονοσειράς, το `$unit` είναι η μονάδα μέτρησης της μεταβλητής, το `$styled` είναι ένα στυλ γραμμών και τα `$start`, `$stop`, `$step` και `$seriesstep` χρησιμοποιούνται για τον καθορισμό της αρχής, του τέλους, του βήματος και του βήματος χρονοσειράς αντίστοιχα.

Το `$date` χρησιμοποιείται για τον υπολογισμό της ημερομηνίας, τα `$x` και `$y` για συντεταγμένες, το `$timelabel` για τον τίτλο στους άξονες, το `$font` για την γραμματοσειρά, το `$fontwidth` για το μήκος της γραμματοσειράς, το `%timestep` για το χρονικό βήμα, το `$deltastep` για την χρονική ολίσθηση και το `%zeromark` για τον καθορισμό του σημείου μηδέν.

Τα `$low`, `$high`, `$minval`, `$maxval` και `$avgval` χρησιμοποιούνται στα διαγράμματα για να δώσουν ακραίες και μέσες τιμές. Τα `$vardate` και `$varvalue` χρησιμοποιούνται για την σημείωση των ακραίως τιμών και της μέσης.

Τα `$xr` και `$yr` είναι οι προηγούμενες τιμές των `$x` και `$y` αντίστοιχα. Το `$timeformat` δίνει την μορφοποίηση του χρόνου και τα `$minvaldate` και `$maxvaldate` δίνουν την ημερομηνία για την ελάχιστη και την μέγιστη τιμή.

```
my $imagewidth=600;
my $imageheight=290;
my $gridX=80;
my $gridY=30;
my $shiftX=-20;
my $shiftY=-10;
my $horizontal=5;
my $vertical=24;
my $XaxisXoffset=5;
my $XaxisYoffset=5;
my $YaxisXoffset=5;
my $YaxisYoffset=8;
my $linemark=3;
my $tolerance=1e-10;
my $spotlength=5;
my $pix=5;
my $DATEFORMAT="YYYYMMDDHH24MISS";
my $TIMEUPLEGEND="%Y/%m/%d %H:%M";
my $TIMERIGHTLEGEND="(%H:%M)";
my $NOMINAVG="rainfall wind direction";
```

Το `$imagewidth` και `$imageheight` είναι το μήκος και το ύψος της εικόνας του διαγράμματος. Το `$gridX` και `$gridY` είναι οι συντεταγμένες του πάνω αριστερά άκρου του ορθογωνίου της γραφικής παράστασης. Τα `$shiftX` και `$shiftY` είναι οι ολισθήσεις της γραφικής παράστασης από το προκαθορισμένο σημείο. Το `$vertical` και το `$horizontal` δίνουν τις κάθετες και τις οριζόντιες διακεκομμένες γραμμές που μπορεί να έχει το διάγραμμα. Οι μεταβλητές `$?axis?offset` καθορίζουν την απόσταση που έχουν οι λεζάντες από τον άξονα. Το `$linemark` δίνει το μήκος της μικρής γραμμής για την υποδιαίρεση που μπαίνει στους άξονες. Το `$tolerance` χρησιμοποιείται για την ισότητα δεκαδικών, το `$spotlength` για το μήκος της κουκίδας τομής και το `$pix` για τα pixels που πρέπει να υπολογιστούν παρακάτω από το `$low` για καλύτερη παρουσίαση. Το `$DATEFORMAT` δίνει

την μορφοποίηση της ημερομηνίας, το \$TIMEUPLEGEND και \$TIMERIGHTLEGEND δίνουν την μορφοποίηση της ημερομηνίας για τις ημερομηνίες που δίνονται πάνω και δεξιά του διαγράμματος και το \$NOMINAVG περιέχει τις μεταβλητές που δεν θα έχουν ελάχιστη και μέση τιμή στο διάγραμμά τους.

```
@arguments=@_ ;

$seriesstep=$arguments[0];
$seriesstep=~s/ly$/i;
$seriesstep="Day" if ($seriesstep eq "Dai");
%timelength=("Ten-minute"=>"-1 day", "Hour"=>"-1 week", "Day"=>"-1 month",
"Month"=>"-1 year");
$timeformat=("Ten-minute"=>"%k", "Hour"=>"%e", "Day"=>"%f",
"Month"=>"%Y");
%timestep=("Ten-minute"=>"10 min", "Hour"=>"1 hour", "Day"=>"1 day",
"Month"=>"1 month");
%zeromark=("Ten-minute"=>"%M", "Hour"=>"%k", "Day"=>"%e",
"Month"=>"%f");
$query="SELECT step_id,name FROM time_step";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare select on time step:\nDB error: $DBI::errstr",$err);
$tsth->execute or
syserr("Can't execute select on time step:\nDB error: $DBI::errstr",$err);
$queries++;
while (@datarow=$tsth->fetchrow_array)
{
    $stepid{$datarow[1]}=$datarow[0];
}
syserr("An error occurred while fetching data:\nDB error: $DBI::errstr",$err)
if ($DBI::err);
```

Αρχικοποίηση του \$seriesstep στο όρισμα που δόθηκε ως τύπος χρονοσειράς και μετατροπή του σε βήμα. Οι μεταβλητές %timelength, \$timeformat, \$timestep και %zeromark ρυθμίζονται ανάλογα για την κάθε περίπτωση χρονικού βήματος. Το hash %stepid με μία εντολή SQL ρυθμίζεται στα χρονικά βήματα που έχει η βάση.

```
if ($timeseries eq "all")
{
    $query="SELECT series_id FROM time_series ".
    "WHERE step_id=$stepid{$arguments[0]}";
    $tsth=$dbh->prepare($query) or
    syserr("Can't prepare series ID:\nDB error: $DBI::errstr",$err);
    $tsth->execute or
    syserr("Can't execute series ID:\nDB error: $DBI::errstr",$err);
    $queries++;
    while (@datarow=$tsth->fetchrow_array)
    {
        push @series,$datarow[0];
    }
}
else
{
    $result=$timeseries;
    @result_array=split /,/, $result;
    for $i (0..$#result_array)
    {
        if ($result_array[$i]=~/\.\./)
        {
            ($first,$last)=split /\.\./,$result_array[$i];
            syserr("Left time series limit $first is greater than $last",$err)
            if ($first>$last);
            for $j ($first..$last)
```

```

    {
        push @series,$j;
    }
}
else
{
    push @series,$result_array[$i];
}
}
}

```

Αν δεν έχει δοθεί η λέξη "all" για να φτιαχτούν διαγράμματα για όλες τις χρονοσειρές, εξετάζεται η λίστα με τις χρονοσειρές χωριζόμενες με κόμμα και την πιθανή παρουσία του τελεστή '..'. Θέτεται κατάλληλα η μεταβλητή @series σε κάθε περίπτωση.

```

for $i (0..$#series)
{
    sysmsg("Drawing graph for $series[$i] $arguments[0] series...\n",0);
    $query="SELECT min(value),max(value),to_char(max(data_date),".
    "$DATEFORMAT') FROM data_proc WHERE series_id=$series[$i]";
    $tsth=$dbh->prepare($query) or
    syserr("Can't prepare max date query:\nDB error: $DBI::errstr",$err);
    $tsth->execute or
    syserr("Can't execute max date query:\nDB error: $DBI::errstr",$err);
    $queries++;
    @columns_array=$tsth->fetchrow_array or
    syserr("Can't fetch results for max data:\nDB error: $DBI::errstr",$err);
    $low=$columns_array[0];
    $high=$columns_array[1];
    $maxdate=$columns_array[2];
    sysmsg("Series low=$low, high=$high and maxdate=$maxdate\n",5);
    syserr("No max date returned from $series[$i] $arguments[0] series",
    $err) unless (defined $maxdate);
    $delta=ParseDateDelta($timelength{$seriesstep});
    $mindate=DateCalc($maxdate,$delta);
    $mindate=~tr/://d;
    sysmsg("Calculated mindate $mindate from maxdate $maxdate\n",5);
}

```

Για την κάθε χρονοσειρά που έχει δοθεί παίρνουμε την ελάχιστη τιμή, την μέγιστη τιμή και την μέγιστη ημερομηνία και την αποθηκεύουμε αντίστοιχα στις μεταβλητές \$low, \$high και \$maxdate. Από το \$maxdate κάνουμε μία ολίσθηση προς τα πίσω και παίρνουμε το \$mindate.

```

$query="SELECT min(value),max(value),avg(value) FROM data_proc ".
"WHERE series_id=$series[$i] AND ".
"data_date>=to_date($mindate,'$DATEFORMAT') AND ".
"data_date<=to_date($maxdate,'$DATEFORMAT')";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare $arguments[0] series min,max query:\nDB error: ".
"$DBI::errstr",$err);
$tsth->execute or
syserr("Can't execute $arguments[0] series min,max query:\nDB error: ".
"$DBI::errstr",$err);
$queries++;
@columns_array=$tsth->fetchrow_array or
syserr("Can't fetch min,max data:\nDB error: $DBI::errstr",$err);
$minval=$columns_array[0];
$maxval=$columns_array[1];
$avgval=$columns_array[2];
unless (defined $minval and defined $maxval and defined $avgval)
{
    syserr("No values presence at this time duration. Ignoring $series[$i] ".
    "time series",0);
    binmode IMG;
}

```

```

print IMG $image->jpeg;
close(IMG) or
syserr("Can't close image file $imagefile:\nSystem error: $!", $err);
next;
}
sysmsg("Minval=$minval, maxval=$maxval and avgval=$avgval\n",5);

```

Επιλέγεται η ελάχιστη, η μέγιστη και η μέση τιμή της χρονοσειράς για το συγκεκριμένο χρονικό παράθυρο. Αν αυτές δεν είναι ορισμένες, τότε σημαίνει ότι δεν έχουμε καθόλου δεδομένα για αυτό το χρονικό παράθυρο και με ένα μήνυμα λάθους τερματίζουμε το πρόγραμμα σώζοντας την εικόνα ως jpg.

```

$query="SELECT to_char(data_date,'$DATEFORMAT') FROM data_proc ".
"WHERE series_id=$series[$i] AND value=$minval AND ".
"data_date>=to_date($mindate,'$DATEFORMAT') AND ".
"data_date<=to_date($maxdate,'$DATEFORMAT')";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare $arguments[0] series min value date query:\n".
"DB error: $DBI::errstr", $err);
$tsth->execute or
syserr("Can't execute $arguments[0] series min value date query:\nDB error: ".
"$DBI::errstr", $err);
$queries++;
@columns_array=$tsth->fetchrow_array or
syserr("Can't fetch min,max data:\nDB error: $DBI::errstr", $err);
$minvaldate=$columns_array[0];

$query="SELECT to_char(data_date,'$DATEFORMAT') FROM data_proc ".
"WHERE series_id=$series[$i] AND value=$maxval AND ".
"data_date>=to_date($mindate,'$DATEFORMAT') AND ".
"data_date<=to_date($maxdate,'$DATEFORMAT')";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare $arguments[0] series max value date query:\n".
"DB error: $DBI::errstr", $err);
$tsth->execute or
syserr("Can't execute $arguments[0] series max value date query:\n".
"DB error: $DBI::errstr", $err);
$queries++;
@columns_array=$tsth->fetchrow_array or
syserr("Can't fetch min value date data:\nDB error: $DBI::errstr", $err);
$maxvaldate=$columns_array[0];
sysmsg("Minvaldate=$minvaldate and maxvaldate=$maxvaldate\n",5);

```

Υπολογίζουμε την ημερομηνία που παρουσιάστηκε η ελάχιστη και η μέγιστη τιμή στο συγκεκριμένο χρονικό παράθυρο.

```

$query="SELECT v.name,v.gr_name,v.unit,s.name,s.gr_name ".
"FROM time_series t,variable v,time_step s ".
"WHERE t.series_id=$series[$i] AND t.variable_id=v.variable_id ".
"AND t.step_id=s.step_id";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare $arguments[0] series info query:\nDB error: ".
"$DBI::errstr", $err);
$tsth->execute or
syserr("Can't execute $arguments[0] series info query:\nDB error: ".
"$DBI::errstr", $err);
$queries++;
@columns_array=$tsth->fetchrow_array or
syserr("Unable to fetch data from $arguments[0] series info:\nDB error: ".
"$DBI::errstr", $err);
%varname=("en"=>$columns_array[0], "gr"=>$columns_array[1]);
$unit=$columns_array[2];
%stepname=("en"=>$columns_array[3], "gr"=>$columns_array[4]);

```

```
sysmsg("Var name=$varname{'en'},grvar name=$varname{'gr'},unit=$unit\n",5);
sysmsg("Step name=$stepname{'en'}, gr step name=$stepname{'gr'}\n",5);
```

Ερώτηση στη βάση συλλογής πληροφοριών για τα ονόματα της μεταβλητής και του χρονικού βήματος που χρησιμοποιεί η συγκεκριμένη χρονοσειρά, καθώς και της μονάδας μέτρησης.

```
$language="en";
$imagefile=lc($varname{$language});
$imagefile=~s/ /_/g;
$imagefile=$IMAGEDIR."/graphs-$language/$imagefile$series[$i].jpg";

open(IMG, ">$imagefile") or
syserr("Can't create image file $imagefile:\nSystem error: $!", $err);

$image=new GD::Image($imagewidth,$imageheight);
```

Η γλώσσα που χρησιμοποιείται είναι τα αγγλικά. Στην συνέχεια φτιάχνεται το όνομα του αρχείου που θα κρατήσει την εικόνα. Αυτό θα είναι το ίδιο με το όνομα της μεταβλητής που μετράει ο αισθητήρας σε πεζά γράμματα. Αν υπάρχει κενό στην περιγραφή της μεταβλητής τότε αυτό αντικαθίσταται από το '_' . Το τελικό όνομα προκύπτει με ένωση του ονόματος της μεταβλητής και του αριθμού της χρονοσειράς.

```
$color{"white"}=$image->colorAllocate(255,255,255);
$color{"black"}=$image->colorAllocate(0,0,0);
$color{"blue"}=$image->colorAllocate(0,0,255);
$color{"red"}=$image->colorAllocate(255,0,0);
$color{"green"}=$image->colorAllocate(0,255,0);

$trans=gdTransparent;
$styled=gdStyled;
$font=gdSmallFont;
$fontwidth=$font->width;
$fontheight=$font->height;
$image->setStyle($color{"black"},$trans,$trans,$trans,$trans);

$image->rectangle($gridX+$shiftX,$gridY+$shiftY,$imagewidth-$gridX+$shiftX,
$imageheight-$gridY+$shiftY,$color{"black"});
```

Δημιουργούνται μερικά χρώματα που θα χρησιμοποιηθούν μετά. Το πρώτο χρώμα που δημιουργήθηκε είναι αυτόματα το φόντο της εικόνας ολόκληρης. Αρχικοποίηση γίνεται και στις μεταβλητές που περιγράφουν την γραμματοσειρά, την διαφάνεια και το στυλ. Ζωγραφίζουμε ένα ορθογώνιο που θα κρατήσει την γραφική παράσταση.

```
$query="SELECT to_char(data_date,'$DATEFORMAT'),value FROM data_proc ".
"WHERE series_id=$series[$i] AND ".
"data_date>=to_date($mindate,'$DATEFORMAT') AND ".
"data_date<=to_date($maxdate,'$DATEFORMAT') ORDER BY data_date";
$tsth=$dbh->prepare($query) or
syserr("Can't prepare $arguments[0] series value query:\n".
"DB error: $DBI::errstr", $err);
$tsth->execute or
syserr("Can't execute $arguments[0] series value query:\nDB error: ".
"$DBI::errstr", $err);
$queries++;
```

Ερώτηση ανάκτηση όλων των δεδομένων που υπάρχουν για την συγκεκριμένη χρονοσειρά στο χρονικό διάστημα που μας ενδιαφέρει.

```
@columns_array=$tsth->fetchrow_array or
syserr("Unable to fetch first value data:\nDB error: $DBI::errstr", $err);
$high=($high+$maxval)/2;
```

```

$result=($low+$minval)/2;
$columns=$gridY+$shiftY+
(($high-$result)*($imageheight-2*$gridY))/($high-$low);
if ($imageheight-$gridY+$shiftY-($gridY+$shiftY+
(($high-$result)*($imageheight-2*$gridY))/($high-$low))<$pix)
{
    $low=$high-
    (($imageheight-2*$gridY+$pix)*($high-$low))/($imageheight-2*$gridY);
}
else
{
    $low=$result;
}

```

Παίρνουμε τα πρώτα αποτελέσματα για να αρχικοποιήσουμε τις μεταβλητές. Το πάνω όριο γίνεται ίσο με το μισό του αθροίσματος της ανώτερης τιμής του χρονικού παραθύρου και της μέγιστης τιμής της χρονοσειράς. Αν η διαφορά του μισού του αθροίσματος του κάτω ορίου με την ελάχιστη τιμή της χρονοσειράς είναι μικρότερη από \$pix, τότε ξανά υπολογίζεται για \$pix παραπάνω από τον οριζόντιο άξονα, αλλιώς παίρνει την τιμή που έδωσε το μισό του αθροίσματος.

```

$start=UnixDate($mindate,"%s");
$stop=UnixDate($maxdate,"%s");
$vardate=UnixDate($columns_array[0],"%s");
$columns_array[1]=$avgval unless (defined $columns_array[1]);
$xp=$imagewidth-$gridX+$shiftX-
(($stop-$vardate)*($imagewidth-2*$gridX))/($stop-$start);
$yp=$gridY+$shiftY+
(($high-$columns_array[1])*($imageheight-2*$gridY))/($high-$low);
$xp=sprintf("%.0f",$xp);
$yp=sprintf("%.0f",$yp);
sysmsg("Start=$start, stop=$stop, vardate=$vardate, xp=$xp, yp=$yp\n",5);

```

Αρχικοποιούνται η ημερομηνία εκκίνησης και λήξης του χρονικού παραθύρου. Το \$vardate παίρνει την τρέχουσα τιμή της ημερομηνίας και το \$columns_array αν δεν έχει οριστεί παίρνει την μέση τιμή. Υπολογίζονται τα προηγούμενα \$x και \$y, τα \$xp και \$yp δηλαδή, στην πρώτη τιμή που διαβάστηκε και οι τιμές στρογγυλοποιούνται.

```

while (@columns_array=$tsth->fetchrow_array)
{
    $vardate=UnixDate($columns_array[0],"%s");
    if (defined $columns_array[1])
    {
        $x=$imagewidth-$gridX+$shiftX-
        (($stop-$vardate)*($imagewidth-2*$gridX))/($stop-$start);
        $y=$gridY+$shiftY+
        (($high-$columns_array[1])*($imageheight-2*$gridY))/($high-$low);
        $x=sprintf("%.0f",$x);
        $y=sprintf("%.0f",$y);
    }
    else
    {
        $x=$xp;
        $y=$yp;
    }
    $image->line($xp,$yp,$x,$y,$color{"green"});
    $xp=$x;
    $yp=$y;
    $lines++;
}
syserr("Can't fetch value data:\nDB error: $DBI::errstr",$err)
if ($DBI::err);

```


Όσο υπάρχουν δεδομένα το \$vardate θα αρχικοποιείται στην τρέχουσα ημερομηνία. Αν έχει οριστεί η τιμή για την συγκεκριμένη ημερομηνία, τότε υπολογίζονται τα \$x και \$y και στρογγυλοποιούνται, αλλιώς παίρνουν την ίδια τιμή με τις προηγούμενες συντεταγμένες. Στο τέλος φέρνουμε την γραμμή και θέτουμε τις προηγούμενες συντεταγμένες στις τρέχουσες.

```
$image->fill($gridX+$shiftX+1,$imageheight-$gridY+$shiftY-1,
$color{"green"});

$start=UnixDate($mindate,"%s");
$stop=UnixDate($maxdate,"%s");
$deltastep=ParseDateDelta($timestep{$arguments[0]});
$j=DateCalc($mindate,$deltastep);
$j=UnixDate($j,"%s");
sysmsg("X axis start=$start, stop=$stop and deltastep=$deltastep\n",5);
```

Γεμίζουμε το μέρος κάτω από τη γραφική παράσταση με το ίδιο πράσινο χρώμα. Αρχικοποιούμε την ημερομηνία εκκίνησης και λήξης, το βήμα και την τρέχουσα θέση \$j.

```
while ($j<=$stop)
{
    $date=ParseDateString("epoch $j");
    $timelabel=UnixDate($date,$timeformat{$arguments[0]});
    $timelabel=~tr/ //d;
    $step=UnixDate($date,$zeromark{$arguments[0]});
    if ($step==0)
    {
        $x=$imagewidth-$gridX+$shiftX-
        (($stop-$j)*($imagewidth-2*$gridX))/($stop-$start);
        $x=sprintf("%.0f",$x);
        $image->line($x,$gridY+$shiftY,$x,$imageheight-$gridY+$shiftY,$styled);
        $image->line($x,$imageheight-$gridY+$shiftY,$x,
        $imageheight-$gridY+$shiftY+$linemark,$color{"black"});
        if ($timelabel%2==0)
        {
            $image->line($x,$gridY+$shiftY,$x,$imageheight-$gridY+$shiftY,
            $color{"red"}) if ($timelabel==0);
            $XaxisXoffset=$timelabel>=10?5:2;
            $image->string($font,$x-$XaxisXoffset,
            $imageheight-$gridY+$shiftY+$XaxisYoffset,$timelabel,$color{"black"});
        }
    }
    $j=DateCalc($date,$deltastep);
    $j=UnixDate($j,"%s");
}
```

Καθώς δεν έχουμε φτάσει στο τέλος, αρχικοποιούμε το \$timelabel και υπολογίζουμε το \$step να είναι το συγκεκριμένο πεδίο του χρόνου που μας δείχνει ότι υπάρχει μηδενισμός και θα πρέπει να τραβήξουμε μία κάθετη διακεκομμένη. Έτσι αν είναι μηδέν το \$step υπολογίζουμε το \$x, το στρογγυλοποιούμε και τραβάμε την διακεκομμένη οριζόντια γραμμή, καθώς και την μικρούλα γραμμούλα στον άξονα, που δείχνει την υποδιαίρεση. Ανά δύο διακεκομμένες γραμμές βάζουμε και ένα αριθμό που δηλώνει τη συγκεκριμένη χρονική στιγμή. Αν το \$timelabel είναι μηδέν, τότε φέρνουμε αντί για διακεκομμένη, κόκκινη γραμμή για να ξεχωρίζει. Στο τέλος υπολογίζεται πάλι το \$j για την επόμενη επανάληψη.

```
$start=$low;
$stop=$high;
$step=($stop-$start)/$horizontal;

sysmsg("Y axis start=$start, stop=$stop and step=$step\n",5);
for ($j=$start;$j<=$stop+$tolerance;$j+=$step)
{
```

```

$j=$stop if ($j>$stop);
$y=$gridY+$shiftY+((($stop-$j)*($imageheight-2*$gridY))/($stop-$start);
$image->line($gridX+$shiftX-$linemark,$y,$gridX+$shiftX,$y,
$color{"black"});
$image->line($gridX+$shiftX,$y,$imagewidth-$gridX+$shiftX,$y,$styled);
$varvalue=sprintf("%04.1f",$j);
$image->string($font,
$gridX+$shiftX-length("$varvalue")*$fontwidth-$YaxisXoffset,
$y-$YaxisYoffset,$varvalue,$color{"black"});
}

```

Στο παραπάνω κομμάτι, αρχικοποιούνται οι μεταβλητές εκκίνησης, τερματισμού και βήματος για την εισαγωγή των οριζόντιων διακεκομμένων γραμμών. Για την τρέχουσα τιμή του \$j υπολογίζονται οι συντεταγμένες του \$y και ζωγραφίζεται στο διάγραμμα η υποδιαίρεση και η διακεκομμένη γραμμή, με την τιμή της που μπαίνει στο πλάι.

```

$timelabel=UnixDate($mindate,$TIMEUPLEGEND);
$image->string($font,$gridX+$shiftX,
$gridY+$shiftY-$fontheight-$XaxisYoffset,$timelabel,$color{"black"});
$timelabel=UnixDate($maxdate,$TIMEUPLEGEND);
$image->string($font,
$imagewidth-$gridX+$shiftX-length($timelabel)*$fontwidth,
$gridY+$shiftY-$fontheight-$XaxisYoffset,$timelabel,$color{"black"});

```

Στο επάνω μέρος, γράφουμε την ελάχιστη ημερομηνία και στα δεξιά την μέγιστη.

```

$font=gdTinyFont;
$fontheight=$font->height;
$fontwidth=$font->width;
$start=UnixDate($mindate,"%s");
$stop=UnixDate($maxdate,"%s");
$vardate=UnixDate($maxvaldate,"%s");
$x=$imagewidth-$gridX+$shiftX-
((($stop-$vardate)*($imagewidth-2*$gridX))/($stop-$start);
$y=$gridY+$shiftY+((($high-$maxval)*($imageheight-2*$gridY))/($high-$low);
$image->dashedLine($gridX+$shiftX,$y,$imagewidth-$gridX+$shiftX,$y,
$color{"blue"});
$image->dashedLine($x,$imageheight-$gridY+$shiftY,$x,$y,$color{"blue"});
$image->arc($x,$y,$spotlength,$spotlength,0,360,$color{"blue"});
$varvalue=sprintf("%04.1f",$maxval);
$date=UnixDate($maxvaldate,$TIMERIGHTLEGEND);
$timelabel="Max: $varvalue $date";
$image->string($font,$imagewidth-$gridX+$shiftX+$YaxisXoffset,
$y-$fontheight/2,$timelabel,$color{"blue"});

unless ($NOMINAVG=~/$varname{'en'}/i)
{
$vardate=UnixDate($minvaldate,"%s");
$x=$imagewidth-$gridX+$shiftX-
((($stop-$vardate)*($imagewidth-2*$gridX))/($stop-$start);
$y=$gridY+$shiftY+((($high-$minval)*($imageheight-2*$gridY))/($high-$low);
$x=sprintf("%.0f",$x);
$y=sprintf("%.0f",$y);
$image->dashedLine($gridX+$shiftX,$y,$imagewidth-$gridX+$shiftX,$y,
$color{"blue"});
$image->dashedLine($x,$imageheight-$gridY+$shiftY,$x,$y,$color{"blue"});
$image->arc($x,$y,$spotlength,$spotlength,0,360,$color{"blue"});
$varvalue=sprintf("%04.1f",$minval);
$date=UnixDate($minvaldate,$TIMERIGHTLEGEND);
$timelabel="Min: $varvalue $date";
$image->string($font,$imagewidth-$gridX+$shiftX+$YaxisXoffset,
$y-$fontheight/2,$timelabel,$color{"blue"});
$y=$gridY+$shiftY+((($high-$avgval)*($imageheight-2*$gridY))/($high-$low);

```

```

$image->dashedLine($gridX+$shiftX,$y,$imagewidth-$gridX+$shiftX,$y,
$color{"blue"});
$varvalue=sprintf("%04.1f",$avgval);
$timelabel="Avg: $varvalue";
$image->string($font,$imagewidth-$gridX+$shiftX+$YaxisXoffset,
$y-$fontheight/2,$timelabel,$color{"blue"});
}

```

Ζωγραφίζουμε και τις λεζάντες για την μέγιστη τιμή. Για την ελάχιστη και τη μέση, την ζωγραφίζουμε μόνο για τις περιπτώσεις των μεταβλητών που δεν ανήκουν στην κατηγορία που έχει οριστεί στην μεταβλητή \$NOMINAVG.

```

$font=gdGiantFont;
$fontheight=$font->height;
$fontwidth=$font->width;
$timelabel="$varname{'en'} ($unit)";
$image->stringUp($font,0,($imageheight+length($timelabel)*$fontwidth)/2+
$shiftY,$timelabel,$color{"black"});
$timelabel=lc($seriesstep);
$timelabel=~s/ten-//i;
$timelabel=$timelabel."s";
$timelabel="Time ($timelabel)";
$image->string($font,($imagewidth-length($timelabel)*$fontwidth)/2+$shiftX,
$imageheight-$fontheight,$timelabel,$color{"black"});

binmode IMG;

print IMG $image->jpeg;

close(IMG) or
syserr("Can't close image file $imagefile:\nSystem error: $!", $err);
}
}

```

Στο τέλος γράφουμε και με μεγάλα γράμματα τους τίτλους που έχουν οι άξονες, σώζουμε την εικόνα σαν jpg και την κλείνουμε.

4.2.2.16 Ρουτίνα δημιουργίας εντολής MakeQuery

Η συνάρτηση MakeQuery, καλείται στο κυρίως πρόγραμμα για να επιστρέψει το query για την κάθε περίπτωση που θα της ζητηθεί.

```
# MakeQuery function. Error level 4.
```

```

sub MakeQuery
{
my ($columns,@columns_array, @result_array, $result, $defcolumns, $query, $i);
my ($answer,$key,%tuple,$j);

($sec[0],$min[0],$hour[0],$mday[0])=gmtime;

```

Η μεταβλητή \$columns, η @columns_array, η @result_array και η \$result είναι γενικής χρήσεως προσωρινές μεταβλητές. Η μεταβλητή \$defcolumns κρατάει τα εξορισμού πεδία που έχει ο πίνακας, το \$query κρατάει την εντολή SQL και το \$i χρησιμοποιείται στους βρόχους επανάληψης. Το \$answer χρησιμοποιείται για την απάντηση του χρήστη σε ερώτηση του προγράμματος, το \$key για κλειδιά hash, το %tuple για την αντιγραφή του γενικού %table και το \$j για επανάληψη σε βρόχους. Η ρουτίνα ξεκινάει αρχικοποιώντας τον χρόνο.

```
if ($command eq "show")
```

```

{
  sysmsg("Preparing show query...\n",0);
  $defcolumns=SelectData("column_name","all_tab_columns","table_name",
    $tablename);
  sysmsg("Columns to display [$defcolumns]: ",0);
  $columns=<STDIN>;
  chomp $columns;

  if ($columns eq "")
  {
    $columns=$defcolumns;
  }
  $query="SELECT $columns FROM $tablename";
  $lines++;
}

```

Αν η εντολή για να εκτελέσει είναι η show για την εμφάνιση εγγραφών ενός πίνακα, εκτελεί το αντίστοιχο SELECT στο View all_tab_columns για να δείξει στον χρήστη τα πεδία που διαθέτει ο πίνακας και να κάνει επιλογή ποια θέλει να εμφανιστούν και με ποια σειρά. Ανάλογα με την είσοδο του χρήστη, δημιουργείται το \$query και επιστρέφεται στο τέλος της ρουτίνας,

```

elseif ($command eq "showall")
{
  sysmsg("Preparing showall query...\n",0);
  $query="SELECT tname from tab";
  $lines++;
}

```

Αν η εντολή είναι "showall", τότε επιστρέφεται η εντολή που φαίνεται, για την εκτύπωση όλων των πινάκων της βάσης.

```

elseif ($command eq "add")
{
  sysmsg("Prepearing add query...\n",0);
  $defcolumns=SelectData("column_name","all_tab_columns","table_name",
    $tablename);
  @columns_array=split /,/, $defcolumns;

  $i=0;
  while (defined $columns_array[$i])
  {
    sysmsg("Enter $columns_array[$i]: ",0);
    $result_array[$i]=<STDIN>;
    chomp $result_array[$i];
    $i++;
  }

  $result=QuoteStrings(@result_array);

  $query="INSERT INTO $tablename($defcolumns) VALUES($result)";
  $lines++;
}

```

Αν η εντολή είναι προσθήκη εγγραφής, τότε παίρνονται όλα τα πεδία που έχει ο πίνακας που θέλει ο χρήστης να κάνει την εγγραφή και για το καθένα πεδίο, του εμφανίζεται περιμένοντας να δώσει είσοδο. Οι τιμές που έδωσε μετά περνούν στο \$query που επιστρέφεται.

```

elseif ($command eq "delete")
{
  sysmsg("Preparing delete query...\n",0);
  sysmsg("Do you want to delete all tuples (yes/NO)? ",0);
  $answer=<STDIN>;

```

```

chomp $answer;
if ($answer =~ /^y/i)
{
    $query="DELETE $tablename";
}
else
{
    $defcolumns=SelectData("column_name","all_tab_columns","table_name",
        $tablename);
    @columns_array=split /,/, $defcolumns;
    sysmsg("Enter $columns_array[0] to delete: ",0);
    $columns=<STDIN>;
    chomp $columns;
    $columns=QuoteStrings($columns);
    $query="DELETE $tablename WHERE $columns_array[0]=$columns";
}
$lines++;
}

```

Η διαγραφή δίνει την δυνατότητα και για διαγραφή όλων των πεδίων, κάτι που δεν είναι καλό και για αυτό η εξορισμού απάντηση στην ερώτηση αυτή είναι το όχι. Μετά ζητάει από τον χρήστη να προσδιορίσει την τιμή του κλειδιού που θέλει να σβήσει και σχηματίζεται η DELETE εντολή SQL.

```

elsif ($command eq "update")
{
    sysmsg("Preparing update query...\n",0);
    $defcolumns=SelectData("column_name","all_tab_columns","table_name",
        $tablename);
    @columns_array=split /,/, $defcolumns;
    sysmsg("Enter $columns_array[0] to update: ",0);
    $columns=<STDIN>;
    chomp $columns;
    $result=SelectData("*",$tablename,$columns_array[0],$columns);
    @result_array=split /,/, $result;

```

Για την ενημέρωση μίας εγγραφής, ζητείται πάλι από τον χρήστη να καθορίσει την εγγραφή που θέλει να ενημερώσει.

```

    $i=1;
    while (defined $columns_array[$i])
    {
        sysmsg("Update $columns_array[$i] [".DBI::neat($result_array[$i])."]":
            ",0);
        $answer=<STDIN>;
        unless ($answer eq "\n")
        {
            chomp $answer;
            $result_array[$i]=QuoteStrings($answer);
        }
        else
        {
            $result=QuoteStrings($result_array[$i]);
            $result_array[$i]=$result;
        }
        $i++;
    }
}

```

Αν δώσει νέα τιμή από αυτή που έχει το πεδίο και του δίνεται στην αγκύλη, τότε αυτή περνάει στον πίνακα @result_array, αλλιώς περνάει η τιμή που ήδη έχει το πεδίο, όταν ο χρήστης δεν επιθυμεί να το αλλάξει και πατήσει enter.

```

    $i=1;

```

```

while (defined $columns_array[$i])
{
  if ((defined $result_array[$i]) and ($result_array[$i] ne ""))
  {
    unless (defined $query)
    {
      $query="UPDATE $tablename SET $columns_array[$i]=$result_array[$i>";
    }
    else
    {
      $query=$query." ,$columns_array[$i]=$result_array[$i>";
    }
  }
  $i++;
}
$query=$query." WHERE $columns_array[0]='$columns'";
$lines++;
}

```

Στο βρόχο while σχηματίζεται η εντολή για ενημέρωση και επιστρέφεται μέσω της \$query.

```

elsif ($command eq "load")
{
  sysmsg("Loading $filename file...\n",0);
  ReadLoadConfig;

  open(LFH,$filename)
  or sysmsg("Could not open $filename:\nSystem error: $!",4);

  while (<LFH>)
  {

```

Για την εντολή φορτώματος ενός αρχείου δεδομένων στη βάση, εκτελούμε πρώτα την ReadLoadConfig για να διαβάσει το αρχείο παραμετροποίησης και να θέσει τις κατάλληλες μεταβλητές. Μετά ανοίγουμε το αρχείο αυτό και όσο υπάρχουν γραμμές εκτελούμε τα παρακάτω.

```

next unless ($_~/\w/);
chomp $_;
sysmsg("Load line: \"$_\"\n",5);
@result_array=split /$delimiter/, $_;
for ($i=0;$i<=$#result_array;$i++)
{
  sysmsg("result_array[$i]=\"$result_array[$i]\"\n",5);
}
foreach $key (keys %table)
{
  foreach $i (keys %{$table{$key}})
  {
    $tuple{$key}{$i}=$table{$key}{$i};
  }
}
}

```

Αγνοούμε οτιδήποτε βρούμε που δεν είναι κανονικός χαρακτήρας. Αφαιρούμε και το enter και εκτυπώνουμε τι ακριβώς έχει διαβαστεί.

```

for ($i=0; $i<=$#result_array;$i++)
{
  $result_array[$i]=~s/^\s+//;
  $result_array[$i]=~s/\s+$//;
  foreach $key (keys %tuple)
  {
    if (exists $tuple{$key}{$i})

```

```

{
  if (${tuple{$key}{$i}} eq "field")
  {
    $tuple{$key}{$i}=QuoteStrings $result_array[$i];
  }
  elsif (${tuple{$key}{$i}} eq "function")
  {
    $tuple{$key}{$i}=$function{$key}."('".$result_array[$i].
    "','$format{$key}')";
  }
  elsif (${tuple{$key}{$i}}=~/#/)
  {
    $tuple{$key}{$i}=~tr/#'//d;
    $tuple{$key}{$i}=$tuple{$key}{$tuple{$key}{$i}};
  }
}
}
}
}

```

Για καθένα από τα πεδία, αφαιρούμε τα κενά που προηγούνται ή έπονται της εγγραφής. Αν υπάρχει εγγραφή για το πεδίο και τη θέση, τότε αυτό παίρνει την τιμή της στήλης \$i όταν έχει για τιμή το "field", παίρνει την συνάρτηση μορφοποίησης όταν έχει τη λέξη "function" και τέλος παίρνει την τιμή του πεδίου που ορίστηκε προηγουμένως από τον τελεστή '#'.

```

foreach $key (keys %tuple)
{
  foreach $i (keys %{$tuple{$key}})
  {
    sysmsg("tuple{$key}{$i}=$tuple{$key}{$i}\n",5);
  }
}

```

Εκτυπώνεται κυρίως για αποσφαλμάτωση τα περιεχόμενα του hash tuple.

```

if ($loadcommand eq "add")
{
  sysmsg("Preparing add command for load query...\n",1);

  for ($i=0;$i<$fieldnumber;$i++)
  {
    $answer="yes";
    $columns="";
    $result="";
    foreach $key (keys %tuple)
    {
      unless (exists $tuple{$key}{$i})
      {
        $answer="no";
      }
    }

    next if ($answer eq "no");
  }
}

```

Αν η εντολή που έχει δηλωθεί είναι για εισαγωγή, τότε η μεταβλητή \$answer χρησιμοποιείται για να δηλώσει την πληρότητα των εγγραφών που μπορούν να εισαχθούν. Έστω και μία εγγραφή να βρεθεί ότι λείπει, το πρόγραμμα συνεχίζει στην επόμενη στήλη.

```

foreach $key (keys %tuple)
{
  next if (($tuple{$key}{$i} eq "'undef'")
  or ($tuple{$key}{$i} eq "''"));
}

```

```

unless ($columns)
{
    $columns="$key";
}
else
{
    $columns=$columns.",$key";
}
unless ($result)
{
    $result="$tuple{$key}{$i}";
}
else
{
    $result=$result.",$tuple{$key}{$i}";
}
}
}

```

Σε αυτό το σημείο έχουμε ορισμένα με τιμές όλα τα πεδία που έχει ο πίνακας και προχωράμε στην κατασκευή των δύο κόμμα χωριζόμενων λιστών για τη μορφοποίηση της εντολής INSERT.

```

$query="INSERT INTO $tablename($columns) VALUES($result)";
sysmsg("Query formed for in load command: $query\n",3);
$sth=$dbh->prepare($query)
or syserr("Can't prepare SQL statement:\nDB error: ".
"$DBI::errstr", 4);

unless ($sth->execute)
{
    if ($DBI::err==1 and $forceinsert eq "yes")
    {
        syserr("Primary key exists, but it will be deleted",0);
    }
}

```

Αν συμβεί λάθος κατά την εκτέλεση της εισαγωγής και έχει οριστεί η εντολή για να εισαχθεί οπωσδήποτε η διαβασμένη τιμή στη βάση, προχωράμε στην διαγραφή της τιμής αυτής.

```

@columns_array=split /,/, $columns;
@result_array=split /,/, $result;
foreach $key (keys %function)
{
    for ($j=0; $j<=$#columns_array; $j++)
    {
        last if ($columns_array[$j] eq $key);
    }
    $answer=join ', ', $result_array[$j], $result_array[$j+1];
    $result_array[$j]=$answer;
    $j++;
    while (defined $result_array[$j+1])
    {
        $result_array[$j]=$result_array[$j+1];
        $j++;
    }
    pop @result_array;
}
}

```

Διαχωρίζονται πάλι τα πεδία και οι τιμές τους. Στις τιμές μόνο θα πρέπει να προσέξουμε τα πεδία που έχουν μορφοποίηση με κάποια συνάρτηση. Ο διαχωρισμός έγινε με βάση το κόμμα και αυτές οι συναρτήσεις έχουν συνήθως κόμμα μέσα τους. Άρα ελέγχουμε με το hash %function που κρατάει αυτά τα πεδία και όσα βρούμε, τα ενώνουμε πάλι πίσω.

```

if ($#columns_array!=$#result_array)
{

```



```

    syserr("Column and data array length mismatch".
    "$#columns_array!=$#result_array",4);
}

```

Ελέγχουμε και εν το μήκος εξακολουθεί να είναι το ίδιο.

```

$defcolumns="";
for ($j=0; $j<=$#columns_array; $j++)
{
    next if ($result_array[$j] eq "");
    unless ($defcolumns)
    {
        $defcolumns="$columns_array[$j]=$result_array[$j]";
    }
    else
    {
        $defcolumns=$defcolumns." and $columns_array[$j]=".
        "$result_array[$j]";
    }
}
$query="DELETE $tablename where $defcolumns";
sysmsg("Extra delete query formed: $query\n",3);
$sth=$dbh->prepare($query)
or syserr("Could not prepare extra delete statement:\n".
"DB error: $DBI::errstr",4);
$sth->execute
or syserr("Could not delete conflicting record",0);
$queryes++;

```

Δημιουργούμε τις ισότητες που χρειάζεται η εντολή DELETE για να δουλέψει και την εκτελούμε αμέσως.

```

sysmsg("Attempting second insertion on duplicated entry\n",3);

$query="INSERT INTO $tablename($columns) VALUES($result)";
sysmsg("Query formed again: $query\n",3);
$sth=$dbh->prepare($query)
or syserr("Unable to prepare SQL statement:\n DB error:".
" $DBI::errstr\n",4);
$sth->execute
or syserr("Unable to insert data for second time. Ignoring".
" data entry...",0);
$queryes++;
}

```

Προσπαθούμε ξανά να εισάγουμε την εγγραφή και αν αποτύχουμε και δεύτερη φορά, την αγνοούμε τελείως.

```

else
{
    syserr("Can't execute SQL statement:\nDB error: ".
"$DBI::errstr", 4);
}
}
$queryes++;
}
}

```

Διαφορετικά αν δεν έχει τεθεί το \$forceinsert από τον χρήστη, εκτυπώνεται το μήνυμα λάθους και το πρόγραμμα τερματίζει.

```

elseif ($loadcommand eq "delete")

```

```

{
  sysmsg("Preparing delete command for load query...\n",4);

  for ($i=0;$i<$fieldnumber;$i++)
  {
    $answer="yes";
    $result="";
    foreach $key (keys %tuple)
    {
      unless (exists $tuple{$key}{$i})
      {
        $answer="no";
      }
    }

    next if ($answer eq "no");

    foreach $key (keys %tuple)
    {
      next if (($tuple{$key}{$i} eq "'undef'") or
        ($tuple{$key}{$i} eq "'')");

      unless ($result)
      {
        $result="$key=$tuple{$key}{$i}";
      }
      else
      {
        $result=$result." and $key=$tuple{$key}{$i}";
      }
    }
    $query="DELETE $tablename WHERE $result";
    sysmsg("Query formed for in load command: $query\n",3);
    $sth=$dbh->prepare($query)
    or syserr("Can't prepare SQL statement:\nDB error: ".
      "$DBI::errstr", 4);

    sysmsg("Executing SQL statement...\n",1);

    $sth->execute
    or syserr("Can't execute SQL statement:\nDB error: ".
      "$DBI::errstr", 4);
    $queries++;
  }
}

```

Στην περίπτωση που η εντολή για εκτέλεση είναι διαγραφή των δεδομένων που υπάρχουν στο αρχείο, ακολουθείται παρόμοια διαδικασία για τον σχηματισμό της σύνταξης της εντολής DELETE και εκτελείται για κάθε γραμμή εγγραφής.

```

else
{
  syserr("Load command \"$loadcommand\" is not implemented",4);
}
$lines++;
}
close(LFH);
$query="SELECT * FROM tab";
}

```

Αν η εντολή για φόρτωση του αρχείου είναι άγνωστη, εκτυπώνεται κατάλληλο μήνυμα λάθους, το αρχείο κλείνει και επιστρέφεται μία χωρίς αποτέλεσμα εντολή.

```

elseif ($command eq "make")
{
    sysmsg("Preparing make command...\n",0);

    TimeSeriesConvert(ucfirst(lc($timestep)));

    $query="SELECT * FROM tab";
}

```

Αν η εντολή είναι "make", τότε καλείται η TimeSeriesConvert με το όρισμα την χρονοσειρά που θέλουμε να ελέγξουμε ή να μετατρέψουμε.

```

elseif ($command eq "execute")
{
    sysmsg("Preparing execute command...\n",0);
    $lines++;
    $query=$execute;
}

```

Αν η εντολή είναι "execute", τότε επιστρέφει κατευθείαν για να εκτελεστεί από το κυρίως πρόγραμμα.

```

elseif ($command eq "generate")
{
    sysmsg("Preparing generate command...\n",0);

    GenerateGraphs(ucfirst(lc($timestep)));

    $query="SELECT * FROM tab";
}

```

Για την περίπτωση της παραγωγής γραφημάτων, καλείται η GenerateGraphs που περιγράφηκε με όρισμα την χρονοσειρά που θέλουμε να της δημιουργηθεί ο γράφος.

```

else
{
    syserr("I don't know what to do \"$command\" command",4);
}
return $query;
}

```

Στην περίπτωση που η εντολή δεν ανήκει σε μία από τις παραπάνω κατηγορίες, εκτυπώνεται ένα μήνυμα λάθους και το πρόγραμμα τερματίζει, αλλιώς επιστρέφει το \$query.

4.2.2.17 Κυρίως πρόγραμμα (Main)

Στο κυρίως πρόγραμμα καλούνται οι διαδικασίες που αναφέρθηκαν για την λειτουργία του προγράμματος.

```

# Main program body. Error level 1.

STDOUT->autoflush(1);
STDERR->autoflush(1);

$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";
$ENV{NLS_LANG}="AMERICAN_AMERICA.EF8ISO8859P7";
$ENV{ORA_NLS33}="$ENV{ORACLE_HOME}/ocommon/nls/admin/data";

```

Ρυθμίζονται η στάνταρ έξοδος και το στάνταρ λάθος να είναι unbuffered. Αρχικοποιούνται μερικές μεταβλητές της Oracle, έτσι ώστε να μην έχει πρόβλημα να τρέχει με κωδικό που δεν κάνει τέτοιες δηλώσεις.

```
chomp $SERVERNAME;
sysmsg("Starting $PROGNAME version $VERSION on $SERVERNAME host...\n",0);
```

```
ReadArguments;
```

```
ReadConfig;
```

Το πρόγραμμα ξεκινάει δίνοντας ένα μήνυμα εκκίνησης, διαβάζοντας τα ορίσματα και το αρχείο παραμετροποίησής του.

```
$lines=0;
$queryes=0;
$dbdebug=0 unless (defined $debug);
$dbverbose="no" unless (defined $verbose);
$dbsavechanges="yes" unless (defined $savechanges);
$dbbatr{PrintError}=(($verbose eq "yes"?1:0);
$dbbatr{AutoCommit}=(($autocommit eq "yes"?1:0);
```

Αρχικοποίηση μεταβλητών και δήλωση άλλων σε μία εξορισμού τιμή, εφόσον δεν έχουν οριστεί.

```
sysmsg("Connecting to $dbservice database...\n",0);
```

```
$dbh=DBI->connect($data_source, $dblogin, $dbpassword, \%dbbatr)
or syserr("Connecting to database failed:\nDB error: $DBI::errstr", 1);
```

```
$sql=MakeQuery;
```

```
sysmsg("Query formed: $sql\n",3);
```

```
$sth=$dbh->prepare($sql)
or syserr("Can't prepare SQL statement:\nDB error: $DBI::errstr", 1);
```

```
sysmsg("Executing SQL statement...\n",1);
```

```
$sth->execute
or syserr("Can't execute SQL statement:\nDB error: $DBI::errstr", 1);
$queryes++;
```

Σύνδεση με τη βάση, εξαγωγή της εντολής που ζήτησε ο χρήστης από την MakeQuery, προετοιμασία και εκτέλεση της εντολής.

```
PrintStatistics("0", "1", "Query");
```

```
if (($command eq "show") or ($command eq "showall") or ((defined $execute) and
($execute=~/^select/i))
```

```
{
  sysmsg("Printing SQL results...\n",0);
```

```
  while (@row=$sth->fetchrow_array)
  {
    $msg="Row: ".DBI::neat_list(@row)." \n";
    sysmsg($msg,1);
  }
```

```
  syserr("Data fetching terminated early by error:\nDB error: ".
"$DBI::errstr", 0) if $DBI::err;
```

```
PrintStatistics("1", "5", "Print");
```

```
PrintStatistics("0", "8", "Summary");
```

```
}
```

Εκτύπωση στατιστικών για την εντολή ή εντολές που εκτελέστηκαν. Αν η εντολή ανήκει σε κάποια κατηγορία που έχουν αποτελέσματα να δείξουν, τότε εκτυπώνονται αυτά καθώς και τα στατιστικά τους.

```
sysmsg("Closing database cursor...\n",1);
$sth->finish
or syserr("Unable to finish cursor:\nDB error: $DBI::errstr", 0);

if (("add delete update load make"=~/$command/i)
    or ((defined $execute) and
        ($execute=~^(insert|update|delete|create|alter)/i)))
{
    if ($ASK4SAVE eq "yes")
    {
        $msg=($savechanges=~/yes/i?"(YES/no)? ":"(yes/NO)? ");
        sysmsg("Save database changes $msg",0);
        $msg=<STDIN>;
        chomp $msg;
        $msg=$savechanges unless ($msg);

        if ($msg=~/^n/i)
        {
            sysmsg("Aborting database changes...\n",0);
            $dbh->rollback
            or syserr("Unable to roll back database changes:\nDB error: ".
                "$DBI::errstr",0);
        }
        else
        {
            sysmsg("Committing database changes...\n",0);
            $dbh->commit
            or syserr("Unable to commit database changes:\nDB error: ".
                "$DBI::errstr",0);
        }
    }
    else
    {
        if ($savechanges=~/^n/i)
        {
            sysmsg("Aborting database changes...\n",0);
            $dbh->rollback or
            syserr("Unable to roll back database changes:\nDB error: $DBI::errstr",0);
        }
        else
        {
            sysmsg("Committing database changes...\n",0);
            $dbh->commit or
            syserr("Unable to commit database changes:\nDB error: $DBI::errstr",0);
        }
    }
}
}
```

Κλείνεται ο database cursor και ο χρήστης ρωτάται αν θέλει να σώσει τις αλλαγές στη βάση.

```
sysmsg("Closing database connection...\n",0);
$dbh->disconnect
or syserr("Disconnect error:\nDB error: $DBI::errstr()", 1);

sysmsg("Program $PROGNAME finished. Exiting.\n",0);

exit 0;
```

Κλείνεται η σύνδεση στη βάση και το πρόγραμμα τερματίζει επιτυχώς.

4.2.3 Πρόγραμμα dbonline

Το πρόγραμμα αυτό έχει τις ίδιες ρουτίνες Usage, sysmsg, syserr, QuoteStrings, RunQuery, ReadArguments, ReadConfig, ReadLoadConfig, SelectData, AngleFromCartesian, PrintStatistics, InsertData, TimeSeriesConvert, MakeQuery με τον aeolus και η μόνη διαφορά του είναι η εστίαση σε αυτόματη ενημέρωση της βάσης δεδομένων.

Δηλαδή για την εισαγωγή και τον έλεγχο δεδομένων, διαβάζει το αρχείο δεδομένων και προσθέτει τις εγγραφές που έχουν μεγαλύτερη ημερομηνία από το max(data_date) που βρίσκεται στη βάση, ενώ ο Αίολος βάζει όλες τις εγγραφές.

4.2.4 CGI Perl scripts

Η υλοποίηση έγινε με την βοήθεια του CGI.pm module της Perl, για την εύκολη παραγωγή με μεθόδους του αντικειμένου φορμών CGI.

4.2.4.1 Επιλογή γλώσσας με το index.cgi

Για την επιλογή της γλώσσας που θα έχουν οι σελίδες, έχει δημιουργηθεί το index.cgi που επιλέγει με βάση το όνομα που έχει ο υπολογιστής και βρίσκεται μέσα σε κάθε κατάλογο:

```
#!/usr/bin/perl -w
```

```
use CGI;
```

Δήλωση του μονοπατιού που βρίσκεται ο διερμηνέας της Perl και της χρήσης του CGI module.

```
my ($query, $clienthost, $dummy);
```

```
$query=CGI::new;
```

Το \$query χρησιμοποιείται για το αντικείμενο του CGI δομοστοιχείου, το \$clienthost είναι μεταβλητή για το όνομα του υπολογιστή που ζητά να δει την σελίδα και το \$dummy είναι προσωρινή μεταβλητή για τον διαχωρισμό του ονόματος του υπολογιστή όπως γυρνάει από την εντολή nslookup. Αρχικοποιούμε το \$query να είναι ένα αντικείμενο τύπου CGI.

```
$clienthost=$query->remote_host;
```

```
$clienthost=`/usr/bin/nslookup $clienthost | grep "Name"`;
```

```
($dummy, $clienthost)=split /:/, $clienthost;
```

```
$clienthost=~s/^\s+//;
```

```
$clienthost=~s/\s+$//;
```

Αναθέτουμε στο \$clienthost την IP του μηχανήματος που ζητάει να δει μία σελίδα. Εκτελούμε την εντολή nslookup για να πάρουμε το όνομα του μηχανήματος στο DNS και επιλέγουμε την γραμμή που επιστρέφει το πρόγραμμα δίνοντας το όνομα μετά το 'Name:'. Χωρίζουμε τις τιμές με βάση το ':' και αφαιρούμε τα κενά που προηγούνται ή έπονται του ονόματος.

```
if ($clienthost=~/\.gr$/i)
```

```
{
```

```
    print $query->redirect('index-gr.shtml');
```

```

}
else
{
    print $query->redirect('index-en.shtml');
}

```

Αν το όνομα του υπολογιστή τελειώνει σε '.gr', τότε κάνουμε ανακατεύθυνση στις ελληνικές σελίδες από το index-gr.shtml, ενώ σε κάθε άλλη περίπτωση δίνουμε τις αγγλικές σελίδες που ξεκινούν από το index-en.shtml.

4.2.4.2 Επιλογή ιστορικών δεδομένων με histdata.cgi

Για την επιλογή των ιστορικών δεδομένων, εκτελείται το histdata.cgi με την παράμετρο language ρυθμισμένη στο "gr" ή στο "en", όταν καλείται από το index-gr.shtml ή από το index-en.shtml αντίστοιχα:

```

#!/usr/bin/perl

use CGI;

my ($query, $language, $msgfile, $menuitem, $message);
my $WEB_DIR="/home/httpd/html/newmeteo/histdata";

```

Δήλωση του διερμηνέα της Perl και της χρήσης του δομοστοιχείου CGI. Το \$query είναι για το αντικείμενο του δομοστοιχείου που χρησιμοποιείται, η μεταβλητή \$language είναι για την γλώσσα, το \$msgfile κρατάει το όνομα του αρχείου από το οποίο θα εκτυπώσει το μήνυμα στον χρήστη, το \$message χρησιμοποιείται για την εκτύπωση μηνυμάτων στο χρήστη και το \$WEB_DIR κρατάει όλο το μονοπάτι μέχρι τον κατάλογο με τα ιστορικά δεδομένα.

```

sub syserr
{
    if (defined $_[1])
    {
        unless ($_[1]==0)
        {
            print "ERROR($_[1]): $_[0]\n";
            exit $_[1];
        }
        else
        {
            print "WARNING: $_[0]\n";
        }
    }
    else
    {
        print "Usage: syserr(<message string>, <exit status>)<br>\n";
        print "Exit status:<br>\n";
        print "0\t\tprints a warning and continues script execution<br>\n";
        print "n\t\tprints the error message and exits with the nonzero n status\n";
        exit -1;
    }
}

```

Συνάρτηση για την εκτύπωση μηνυμάτων λάθους.

```

$query=new CGI;

print $query->header;

```

```
$language=$query->param('language');
$msgfile=$WEB_DIR."/message1-$language.html";
```

Δημιουργούμε ένα νέο στιγμιότυπο του αντικειμένου CGI. Εκτυπώνουμε τις κατάλληλες επικεφαλίδες και αρχικοποιούμε την γλώσσα και το όνομα του αρχείου με το μήνυμα.

```
open(MSGFH, $msgfile)
or syserr("Could not open message file $msgfile: $!\n",1);

while (<MSGFH>)
{
    print $_;
}

close(MSGFH)
or syserr("Unable to close message file $msgfile: $!\n",1);
```

Ανοίγουμε το αρχείο με το μήνυμα και το εκτυπώνουμε στον χρήστη.

```
print $query->startform(-method=>POST, -action=>"/cgi-
bin/newmeteo/histdata1.cgi");

print $query->hidden(-name=>"language", -default=>$language);

print '<DIV ALIGN="center">';

$message=($language eq "gr"?"Αίτηση χορήγησης Ιστορικών Δεδομένων":
"Historical Data application");

print $query->submit(-name=>"proceed", -value=>$message);

print '</DIV>';

print $query->endform;
```

Στο τέλος βάζουμε και ένα κουμπάκι μέσα σε μία φόρμα που μας πάει στο πρώτο βήμα των ιστορικών δεδομένων.

4.2.4.3 Πρώτο βήμα ιστορικών δεδομένων με histdata1.cgi

Σε αυτό το βήμα, ο χρήστης θα κληθεί να επιλέξει τις μεταβλητές, το χρονικό βήμα και το χρονικό διάστημα που τον ενδιαφέρει:

```
#!/usr/bin/perl

use CGI;
use DBI;
```

Δήλωση του μονοπατιού που βρίσκεται ο διερμηνέας της Perl και γίνεται χρήση των δομοστοιχείων CGI και DBI.

```
my ($query, $language, $htmlfile, $menuitem, $msg);
my $FORMDATA="<!--FORMDATA-->";
my $INCLUDE="#include";
my $WEB_DIR="/home/httpd/html/newmeteo";
my ($data_source,$dbservice,$dblogin,$dbpassword);
my $configfile="/etc/aeolus/cgi.conf";
my %dbattr=(PrintError=>0, RaiseError=>0, AutoCommit=>0);
```



```
my ($dbh, $sth, $sql, @row, @list, %label, $fromdate, $todate);
my $DATEFORMAT="HH24:MI DD/MM/YYYY";
```

Το `$query` χρησιμοποιείται για το στιγμιότυπο του αντικειμένου CGI, το `$language` για την δήλωση της γλώσσας, το `$htmlfile` κρατάει το μονοπάτι του αρχείου που έχει τον κώδικα HTML έτοιμο για εμφάνιση της σελίδας που θα φτιαχτεί, στο `$menuitem` σχηματίζεται η συμβολοσειρά του βήματος που βρισκόμαστε και το `$msg` χρησιμοποιείται για την εκτύπωση μηνυμάτων στον χρήστη. Το `$FORMDATA` κρατάει την συμβολοσειρά που υπάρχει στο αρχείο HTML, για να δηλώσει το σημείο στο οποίο θα μπει η φόρμα με τις μεταβλητές και το `$INCLUDE` δηλώνει τα Server Side Includes (SSI) που θα τα αντικαταστήσει το ίδιο το CGI. Ο κατάλογος που βρίσκονται οι σελίδες δηλώνεται στη μεταβλητή `$WEB_DIR` και τα `$data_source`, `$dbservice`, `$dblogin` και `$dbpassword` χρησιμοποιούνται για την σύνδεση στη βάση. Το αρχείο διαμόρφωσης για όλα τα CGI scripts που έχουν δημιουργηθεί, βρίσκεται μέσα στο `/etc/aeolus/cgi.conf`. Το `hash %dbattr` χρησιμοποιείται για την δήλωση των παραμέτρων της συνόδου με την βάση. Το `$dbh` είναι ο χειριστής της σύνδεσης με τη βάση δεδομένων, το `$sth` είναι ο χειριστής των ερωτήσεων στη βάση, το `$sql` είναι η συμβολοσειρά με την εντολή SQL, το `@row` είναι ο πίνακας που καταγράφει τις επιστρεφόμενες τιμές από την εκτέλεση μίας ερώτησης στη βάση, ο πίνακας `@list` κρατάει τιμές για τα χρονικά βήματα, το `%label` τα ονόματα των βημάτων και το `$fromdate` και `$todate` την μέγιστη και ελάχιστη ημερομηνία, για τα οποία έχουμε δεδομένα στη βάση. Η γενική μεταβλητή `$DATAFORMAT`, φυλάει την μορφή που θα απεικονίζεται η ημερομηνία.

```
sub syserr
{
  if (defined $_[1])
  {
    unless ($_[1]==0)
    {
      print "ERROR($_[1]): $_[0]\n";
      exit $_[1];
    }
    else
    {
      print "WARNING: $_[0]\n";
    }
  }
  else
  {
    print "Usage: syserr(<message string>, <exit status>)\n";
    print "Exit status:\n";
    print "0\t\tprints a warning and continues script execution\n";
    print "n\t\tprints the error message and exits with the nonzero n status\n";
    exit -1;
  }
}
```

Ρουτίνα για τον χειρισμό λαθών.

```
# ReadConfig function. Error level 2
```

```
sub ReadConfig
{
  my $SERVICE="DBService";
  my $LOGIN="DBLogin";
  my $PASSWORD="DBPassword";
  my ($keyword);
  my $confforceinsert;

  if (-e $configfile)
  {
    if (-r $configfile)
    {
```

```

open(CFH,$configfile)
  or syserr("Couldn't read from $configfile:\nSystem error: $!",2);

while (<CFH>)
{
  if ($_ =~ /^#/ )
  {
    next;
  }
  elsif ($_ =~ /\w/)
  {
    if ($_ =~ /$SERVICE/i)
    {
      ($keyword, $dbservice)=split(/=/, $_);
      chomp $dbservice;
      $dbservice=~tr/"'"/d;
    }
    elsif ($_ =~ /$LOGIN/i)
    {
      ($keyword, $dblogin)=split /=/, $_;
      chomp $dblogin;
      $dblogin=~tr/"'"/d;
    }
    elsif ($_ =~ /$PASSWORD/i)
    {
      ($keyword, $dbpassword)=split /=/, $_;
      chomp $dbpassword;
      $dbpassword=~tr/"'"/d;
    }
    else
    {
      syserr("Unsupported configuration line:\n$_",2);
    }
  }
}
}
else
{
  syserr("Config file $configfile exists, but it's unreadable", 2);
}
}
else
{
  syserr("Configuration file $configfile does not exist", 2);
}
close(CFH);
$data_source="DBI:Oracle:$dbservice";
}

```

Ρουτίνα που διαβάζει το αρχείο διαμόρφωσης cgi.conf. Είναι υποσύνολο της αντίστοιχης ρουτίνας του Αίολου.

```

$query=new CGI;

print $query->header;

```

Δημιουργούμε ένα νέο στιγμιότυπο του αντικειμένου CGI και εκτυπώνουμε τις στάνταρ επικεφαλίδες που προηγούνται των αποτελεσμάτων ενός CGI script.

```

$language=$query->param("language");
$htmlfile="$WEB_DIR/histdata/skeleton-$language.shtml";
$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";

```

```
$ENV{NLS_LANG}="AMERICAN_AMERICA.EL8ISO8859P7";
$ENV{ORA_NLS33}="$ENV{ORACLE_HOME}/ocommon/nls/admin/data";
```

Αρχικοποιείται η γλώσσα που θα χρησιμοποιήσουμε και το αρχείο που έχει τον σκελετό της σελίδας. Οι δηλώσεις περιβάλλοντος για την Oracle, γίνονται μέσα στο πρόγραμμα, γιατί ο χρήστης nobody με τον οποίο τρέχει το CGI script από το WWW, δεν το έχει σωστά ρυθμισμένο.

```
ReadConfig;
```

```
$dbh=DBI->connect($data_source, "$dblogin", "$dbpassword", \%dbattr)
or syserr("Connecting to database failed:<BR>Database Error: $DBI::errstr",1);
```

Διαβάζει το αρχείο διαμόρφωσης και συνδέεται στη βάση δεδομένων.

```
open(HTMLFH,$htmlfile)
or syserr("Unable to open html file $htmlfile:<BR>System error: $!",1);
```

```
while (<HTMLFH>)
{
```

Ανοίγουμε το αρχείο με το σκελετό της σελίδας για να το εκτυπώσουμε.

```
if ($_~/ $INCLUDE/i)
{
    $_~/s/\s*!--\s*#include\s+virtual=//i;
    $_~/s/\s*-->\s*//i;
    $_~/tr/'"//d;
```

Αν συναντήσουμε SSI, θα πρέπει να τα μεταφράσουμε εμείς οι ίδιοι. Έτσι αφαιρούμε τα αναγνωριστικά του include για να πάρουμε το μονοπάτι που βρίσκεται το αρχείο που θέλουμε να συμπεριλάβουμε.

```
if ($_~/cgi-bin/)
{
    $menuitem=$_;
    $menuitem=~s/.*\?//;
    $menuitem=~s/.*menuitem=//;
    $menuitem=~s/&.*//;
    open(MENUFH, "$WEB_DIR"/histdata/steps-$language.html")
    or syserr("Unable to open menu file $_:<BR> System error: $!",2);
```

Αν το μονοπάτι περιέχει τον κατάλογο /cgi-bin, τότε θα είναι για την ρύθμιση του μενού και θα πρέπει να ανοίξουμε το αρχείο και να αντικαταστήσουμε το μενού με το τρέχον βήμα, με το δεύτερο εικονίδιο που φαίνεται να είναι πατημένο. Στο \$menuitem βάζουμε το όνομα του βήματος που βρισκόμαστε από την παράμετρο που περνιέται στο CGI script που ρυθμίζει το μενού.

```
while (<MENUFH>)
{
    if ($_~/ $menuitem/i)
    {
        $_~/s#src="/buttons/histdata-
$language/{menuitem}1.jpg"#src="/buttons/histdata-
$language/{menuitem}2.jpg"#i;
    }
    print $_;
}

close(MENUFH)
or syserr("Unable to close steps file $_: $!\n",1);
}
```

Καθώς υπάρχουν γραμμές, αν αυτές περιέχουν την εικόνα με το τρέχον βήμα, το αντικαθιστούμε με το δεύτερο εικονίδιο που έχει, αλλιώς εκτυπώνουμε την γραμμή όπως είναι και κλείνουμε το αρχείο.

```

else
{
    open(INCFH, "$WEB_DIR"/.$_)
    or syserr("Unable to open include file $WEB_DIR/$_:<BR>System error:
$!",2);

    while (<INCFH>)
    {
        print $_;
    }

    close(INCFH)
    or syserr("Unable to close file $WEB_DIR/$_:<BR>System error: $!",2);
}
next;
}

```

Διαφορετικά αν έχουμε ένα απλό include αρχείου, το ανοίγουμε και εκτυπώνουμε τα περιεχόμενά του στο συγκεκριμένο σημείο που βρήκαμε την αναφορά.

```

elsif ($_ =~ /$FORMDATA/i)
{
    print $query->startform(-method=>POST,
        -action=>"/cgi-bin/newmeteo/histdata2.cgi");

    print $query->hidden(-name=>"language", -default=>$language);
}

```

Διαφορετικά αν βρούμε αναφορά για FORMDATA, ξεκινούμε την εισαγωγή της φόρμας, με μέθοδο post, όπου με το πάτημα του κουμπιού submit θα μας πάει στο CGI histdata2.cgi που θα αναλάβει το επόμενο βήμα. Η γλώσσα που περάστηκε στην αρχή ξεκινώντας το πρώτο βήμα, περνιέται σαν κρυφή παράμετρος.

```

$msg=($language eq "gr"? "1. Μεταβλητές": "1. Variables");
$msg=$query->b($msg);
print $query->h2($msg);
print $query->br;
$msg=($language eq "gr"? "Προσδιορίστε τις μεταβλητές που σας ενδιαφέρουν":
"Define your desired variables:");
print $query->h4($msg);

```

Εκτυπώνονται μηνύματα ανάλογα με την γλώσσα που έχει επιλεγεί, για τις μεταβλητές.

```

$sql=($language eq "gr"? "SELECT variable_id,gr_name from variable":
"SELECT variable_id,name from variable");
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare variable query:<BR>Database error:
$DBI::errstr",
1);

$sth->execute
or syserror("Could not execute variable query:<BR>".
"Database Error: $DBI::errstr",1);

print "<BLOCKQUOTE>\n";
while (@row=$sth->fetchrow_array)
{
    $msg=$query->checkbox(-name=>"variable$row[0]",
        -value=>"$row[0]",

```

```

        -label=>"$row[1]");
        print "$msg<BR>\n";
    }
    syserr("Data variable fetching terminated early by error:<BR>Database
error:".
    "$DBI::errstr",0) if ($DBI::err);

    print "</BLOCKQUOTE>\n";

```

Σχηματίζουμε την εντολή SQL που θα εκτελεστεί στη βάση για να μας φέρει όλες τις μεταβλητές στην μεταβλητή \$sql. Την εκτελούμε στην βάση και τα εκτυπώνουμε το αποτέλεσμα με την μορφή checkbox.

```

$msg=($language eq "gr"?"Προσδιορίστε το χρονικό βήμα που χρειάζεστε:"
"Define the desired time step:");

print $query->h4($msg);

$sql=($language eq "gr"? "SELECT step_id,gr_name from time_step":
"SELECT step_id,name from time_step");
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare time step query:<BR>".
"Database error: $DBI::errstr",
1);

$sth->execute
or syserr("Could not execute step query:<BR>".
"Database Error: $DBI::errstr",1);

print "<BLOCKQUOTE>\n";
while (@row=$sth->fetchrow_array)
{
    push @list, $row[0];
    $label{$row[0]}="$row[1]";
}
syserr("Data step fetching terminated early by error:<BR>Database error: ".
"$DBI::errstr",0) if ($DBI::err);

print $query->radio_group(-name=>"timestep", -values=>\@list,
-default=>$list[1], -linebreak=>"true", -labels=>\%label);

print "</BLOCKQUOTE>\n";

```

Εκτυπώνεται ένα μήνυμα για τα χρονικά βήματα που ενδιαφέρουν τον χρήστη. Προετοιμάζεται η SQL εντολή ανάλογα με την γλώσσα που έχει επιλεγεί και εκτελείται. Τα αποτελέσματα παρουσιάζονται στον χρήστη με radio buttons.

```

$msg=($language eq "gr"? "Προσδιορίστε το χρονικό διάστημα που σας ".
"ενδιαφέρει:": "Define your desired time duration:");

print $query->h4($msg);
$sql="SELECT to_char(min(data_date), '$DATEFORMAT') from data_proc";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare min data query:<BR>".
"Database error: $DBI::errstr",1);
$sth->execute
or syserr("Could not execute min data query:<BR>".
"Database Error: $DBI::errstr",1);
@row=$sth->fetchrow_array;
$fromdate=$row[0];
$sql="SELECT to_char(max(data_date), '$DATEFORMAT') from data_proc";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare max data query:<BR>".

```

```

"Database error: $DBI::errstr",1);
$sth->execute
or syserr("Could not execute max data query:<BR>".
"Database Error: $DBI::errstr",1);
@row=$sth->fetchrow_array;
$todate=$row[0];

print "<BLOCKQUOTE>\n";
$msg=($language eq "gr"? "Από &nbsp;": "From &nbsp;");
print $msg;
print $query->textfield(-name=>"fromdate", -default=>"$fromdate",
-size=>16, -maxlength=>16, -override=>1);
$msg=($language eq "gr"? "&nbsp; μέχρι &nbsp;": "&nbsp; to &nbsp;");
print $msg;
print $query->textfield(-name=>"todate", -default=>"$todate",
-size=>16, -maxlength=>16, -override=>1);
print "</BLOCKQUOTE>\n";

```

Στο τέλος εκτυπώνεται ένα μήνυμα για την επιλογή του χρονικού βήματος που ενδιαφέρει τον χρήστη. Ετοιμάζονται δύο εντολές για να βρεθεί στη βάση η ελάχιστη και η μέγιστη ημερομηνία. Τα αποτελέσματα παρουσιάζονται με μορφή text field, έτσι ώστε ο χρήστης να μπορεί να αλλάξει την ημερομηνία σύμφωνα με την δική του προτίμηση.

```

print $query->br;
$msg=($language eq "gr"? "Προχωρήστε στους αισθητήρες": "Continue to
sensors");
print $query->submit(-name=>"proceed", -value=>$msg);
print "&nbsp;";
$msg=($language eq "gr"? "Καθαρισμός φόρμας": "Clear form");
print $query->reset($msg);

$sth->finish
or syserr("Unable to finish cursor:<BR>Database error: $DBI::errstr", 0);

print $query->endform;
next;
}

```

Στο τέλος της φόρμας, μπαίνουν και δύο κουμπιά για να προχωρήσει ο χρήστης στο επόμενο βήμα ή να καθαρίσει τις επιλογές που έχει βάλει.

```

print $_;
}

close(HTMLFH)
or syserr("Unable to close html file $htmlfile:<BR>System error: $!",1);

$dbh->disconnect
or syserr("Disconnect error:<BR>Database error: $DBI::errstr()", 1);

exit 0;

```

Σε κάθε άλλη περίπτωση εκτυπώνεται το αρχείο με το σκελετό της σελίδας όπως είναι. Όταν τελειώσει η επεξεργασία του αρχείου, το κλείνουμε και πριν το script τερματίσει, γίνεται αποσύνδεση από την βάση δεδομένων.

4.2.4.4 Δεύτερο βήμα ιστορικών δεδομένων histdata2.cgi

Στο δεύτερο βήμα παρουσιάζονται οι αισθητήρες που υπάρχουν στον βαθμό, ανάλογα με τις μεταβλητές που έχει επιλέξει ο χρήστης στο προηγούμενο βήμα.

```
#!/usr/bin/perl
```

```
use CGI;
use DBI;
```

Δήλωση του διεργμαίου της γλώσσας και χρήση των δομοστοιχείων CGI και DBI.

```
my ($query, $language, $htmlfile, $menuitem, $msg, @variable_array, $timestep);
my $FORMDATA="<!--FORMDATA-->";
my $INCLUDE="#include";
my $WEB_DIR="/home/httpd/html/newmeteo";
my ($data_source, $dbservice, $dblogin, $dbpassword);
my $configfile="/etc/aeolus/cgi.conf";
my %dbattr=(PrintError=>0, RaiseError=>0, AutoCommit=>0);
my ($dbh, $sth, $sql, @row, $fromdate, $todate, $variable, $i);
my ($notavailh, $notavailrow, %variable_hash);
my $DATEFORMAT="HH24:MI DD/MM/YYYY";
my $type=2;
```

Το `$query` είναι για την δήλωση του στιγμιότυπου του αντικειμένου CGI, το `$language` για τη γλώσσα που θα χρησιμοποιηθεί, το `$htmlfile` είναι το αρχείο HTML με το σκελετό της σελίδας, το `$menuitem` είναι το τρέχον βήμα που βρισκόμαστε, το `$msg` είναι τα μηνύματα για τον χρήστη, το `@variable_array` είναι ένας πίνακας για τις μεταβλητές και το `$timestep` είναι για το χρονικό βήμα. Το `$FORMDATA` είναι το αναγνωριστικό που δηλώνει την έναρξη εισαγωγής της φόρμας σε αυτό το σημείο, το `$INCLUDE` για να δηλώσει το `#include` και το `$WEB_DIR` έχει το μονοπάτι με την ρίζα των ιστοσελίδων. Τα `$data_source`, `$dbservice`, `$dblogin` και `$dbpassword` χρησιμοποιούνται ως παράμετροι σύνδεσης στη βάση. Η μεταβλητή `$configfile` έχει το όνομα του αρχείου διαμόρφωσης και το `%dbattr` παραμέτρους που καθορίζουν την συμπεριφορά της βάσης κατά την εκτέλεση εντολών. Το `$dbh` είναι ο χειριστής της βάσης δεδομένων, το `$sth` είναι ο χειριστής των εντολών της βάσης, το `$sql` είναι η μεταβλητή που κρατάει την εντολή που θα εκτελεστεί, το `@row` κρατάει τα αποτελέσματα της εκτέλεσης, το `$fromdate` και το `$todate` κρατάν την ελάχιστη και μέγιστη ημερομηνία που έδωσε ο χρήστης, το `$variable` τον ακεραίο αναγνωριστικό αριθμό της μεταβλητής και το `$i` είναι βοηθητική μεταβλητή για τους βρόχους. Το `$notavailh` είναι χειριστής εντολής για την περίπτωση του διαστήματος μη διαθεσιμότητας ενός αισθητήρα, το `$notavailrow` είναι για τα αποτελέσματα της εκτέλεσης και το `%variable_hash` κρατάει τα αναγνωριστικά των μεταβλητών με κλειδιά το όνομά τους.

```
sub syserr
{
  if (defined $_[1])
  {
    unless ($_[1]==0)
    {
      print "ERROR($_[1]): $_[0]\n";
      exit $_[1];
    }
    else
    {
      print "WARNING: $_[0]\n";
    }
  }
  else
  {
    print "Usage: syserr(<message string>, <exit status>)\n";
    print "Exit status:\n";
    print "0\t\tprints a warning and continues script execution\n";
    print "n\t\tprints the error message and exits with the nonzero n status\n";
  }
}
```

```

    exit -1;
}
}

```

Ρουτίνα χειρισμού λαθών.

```

sub ReadConfig
{
    my $SERVICE="DBService";
    my $LOGIN="DBLogin";
    my $PASSWORD="DBPassword";
    my $keyword;
    my $confforceinsert;

    if (-e $configfile)
    {
        if (-r $configfile)
        {
            open(CFH,$configfile)
                or syserr("Couldn't read from $configfile:\nSystem error: $!",2);

            while (<CFH>)
            {
                if ($_ =~ /^#/)
                {
                    next;
                }
                elsif ($_ =~ /\w/)
                {
                    if ($_ =~ /$SERVICE/i)
                    {
                        ($keyword, $dbservice)=split(/=/, $_);
                        chomp $dbservice;
                        $dbservice=~tr/"'"/d;
                    }
                    elsif ($_ =~ /$LOGIN/i)
                    {
                        ($keyword, $dblogin)=split /=/, $_;
                        chomp $dblogin;
                        $dblogin=~tr/"'"/d;
                    }
                    elsif ($_ =~ /$PASSWORD/i)
                    {
                        ($keyword, $dbpassword)=split /=/, $_;
                        chomp $dbpassword;
                        $dbpassword=~tr/"'"/d;
                    }
                    else
                    {
                        syserr("Unsupported configuration line:\n$_",2);
                    }
                }
            }
        }
        else
        {
            syserr("Config file $configfile exists, but it's unreadable", 2);
        }
    }
    else
    {
        syserr("Configuration file $configfile does not exist", 2);
    }
}

```



```
close(CFH);
$data_source="DBI:Oracle:$dbservice";
}
```

Ρουτίνα διαβάσματος αρχείου διαμόρφωσης.

```
$query=new CGI;

print $query->header;
```

Δημιουργείται ένα καινούργιο στιγμιότυπο του αντικειμένου CGI και εκτυπώνονται οι επικεφαλίδες στην έξοδο.

```
$language=$query->param("language");
$timestep=$query->param("timestep");
$fromdate=$query->param("fromdate");
$todate=$query->param("todate");
$htmlfile="$WEB_DIR/histdata/skeleton-$language.shtml";
```

```
$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";
$ENV{NLS_LANG}="AMERICAN_AMERICA.EL8ISO8859P7";
$ENV{ORA_NLS33}="/home/oracle/ora816/ocommon/nls/admin/data";
```

Διαβάζονται οι παράμετροι που περάστηκαν από το προηγούμενο βήμα με τη μέθοδο POST και αποθηκεύονται στις μεταβλητές \$language, \$timestep, \$fromdate και \$todate. Εκτός από την γλώσσα, οι υπόλοιπες δεν θα χρειαστούν σε αυτό το βήμα, αλλά θα διαδοθούν στο επόμενο. Γίνεται και η ρύθμιση των μεταβλητών περιβάλλοντος για την σωστή λειτουργία της σύνδεσης στην βάση.

```
ReadConfig;

$dbh=DBI->connect($data_source, "$dblogin", "$dbpassword", \%dbattr)
or syserr("Connecting to database failed:<BR>Database Error: $DBI::errstr",1);

$menuitem=($language eq "gr"? "ais8hthres":"sensors");

open(HTMLFH,$htmlfile)
or syserr("Unable to open html file $htmlfile:<BR>System error: $!",1);

while (<HTMLFH>)
{
```

Διαβάζεται το αρχείο διαμόρφωσης με την κλήση της ρουτίνας ReadConfig. Γίνεται σύνδεση στη βάση και ρυθμίζεται το \$menuitem στο συγκεκριμένο βήμα. Ανοίγουμε το αρχείο με τον HTML κώδικα της σελίδας που θα παρουσιάσουμε και κάνουμε τις παρακάτω αλλαγές, όσο υπάρχουν γραμμές.

```
if ($_ =~ /$INCLUDE/i)
{
    $_ =~ s/\s*<!--\s*#include\s+virtual=//i;
    $_ =~ s/\s*-->\s*//i;
    $_ =~ tr/' ' //d;
```

Αν συναντήσουμε γραμμή που να περιέχει include, τότε θα πρέπει να φορτώσουμε εμείς το αντίστοιχο αρχείο. Έτσι αφαιρούμε τα αναγνωριστικά της σύνταξης του include και κρατάμε μόνο το μονοπάτι.

```
if ($_ =~ /cgi-bin/)
{
    open(MENUFH, "$WEB_DIR"./histdata/steps-$language.html")
```

```

or syserr("Unable to open menu file $_:<BR> System error: $!",2);

while (<MENUFH>)
{
  if ($_=~/$menuitem/i)
  {
    $_=~s#src="/buttons/histdata-
$language/${menuitem}1.jpg"#src="/buttons/histdata-
$language/${menuitem}2.jpg"#i;
  }
  print $_;
}

close(MENUFH)
or syserr("Unable to close steps file $_: $!\n",1);
}

```

Αν το μονοπάτι ξεκινάει από /cgi-bin τότε θα πρέπει να αλλάξουμε το μενού με την εικόνα 2 του τρέχοντος βήματος.

```

else
{
  open(INCFH, "$WEB_DIR."/"$_")
  or syserr("Unable to open include file $WEB_DIR/$_:<BR>System error:
$!",2);

  while (<INCFH>)
  {
    print $_;
  }

  close(INCFH)
  or syserr("Unable to close file $WEB_DIR/$_:<BR>System error: $!",2);
}
next;
}

```

Διαφορετικά απλώς θα συμπεριλάβουμε το αρχείο που ορίζει το include και θα συνεχίσουμε στην επόμενη γραμμή.

```

elsif ($_=~/$FORMDATA/i)
{
  print $query->startform(-method=>POST,
  -action=>"/cgi-bin/newmeteo/histdata3.cgi");

  print $query->hidden(-name=>"language", -default=>$language);
  print $query->hidden(-name=>"timestep", -default=>$timestep);
  print $query->hidden(-name=>"fromdate", -default=>$fromdate);
  print $query->hidden(-name=>"todate", -default=>$todate);
}

```

Αν συναντήσουμε τη γραμμή που έχει το σημάδι που δηλώνει ότι θα πρέπει να ξεκινήσει η φόρμα, ξεκινάμε μία νέα φόρμα με τη μέθοδο POST να οδηγήσει στο επόμενο βήμα. Μεταφέρουμε τις παραμέτρους του πρώτου βήματος στο τρίτο βήμα.

```

$msg=($language eq "gr"?2. Αισθητήρες:"2. Sensors");
$msg=$query->b($msg);
print $query->h2($msg);
print $query->br;
$msg=($language eq "gr"?"Προσδιορίστε τους αισθητήρες που θέλετε να ".
"πάρτε δεδομένα:": "Define desired sensor to obtain their data:");
print "$msg";

```

```

$sql="SELECT variable_id,name,gr_name from variable";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare variable query:<BR>Database error:
$DBI::errstr",
1);
$sth->execute
or syserror("Could not execute variable query:<BR>".
"Database Error: $DBI::errstr",1);

while (@row=$sth->fetchrow_array)
{
    $variable_array[$row[0]]="variable$row[0]";
    $variable_hash{$row[0]}{"name"}=$row[1];
    $variable_hash{$row[0]}{"gr_name"}=$row[2];
}
syserr("Data variable fetching terminated early by error:<BR>Database
error:".
" $DBI::errstr",0) if ($DBI::err);

```

Εκτυπώνεται ένα μήνυμα στον χρήστη για τις μεταβλητές και εκτελείται στη βάση η αντίστοιχη ερώτηση με βάση την γλώσσα που έχει οριστεί, για να πάρουμε τις μεταβλητές που υπάρχουν στη βάση, με το όνομα και το αναγνωριστικό τους.

```

for ($i=1;$i<=#variable_array;$i++)
{
    next unless ($query->param($variable_array[$i]));

    $variable=$i;
    $msg=($language eq "gr"?$variable_hash{$variable}{"gr_name"}:
    $variable_hash{$variable}{"name"});
    $msg=$query->b($msg);
    print $query->h3($msg);
    print "<BLOCKQUOTE>\n";

    $sql="SELECT s.sensor_id,s.name,s.gr_name,s.firm,s.model,".
    "to_char(s.start_date,'$DATEFORMAT'),to_char(s.end_date,'$DATEFORMAT'),".
    "s.accuracy,s.comments,s.gr_comments,t.series_id FROM sensor s,".
    "time_series t WHERE t.variable_id=$variable and s.sensor_id=t.sensor_id".
    " and t.step_id=$timestep and t.type_id=$type";
    $sth=$dbh->prepare($sql)
    or syserr("Unable to prepare sensor query:<BR>".
    "Database error: $DBI::errstr",1);
    $sth->execute
    or syserr("Could not execute sensor query:<BR>".
    "Database Error: $DBI::errstr",1);
}

```

Για όλες τις μεταβλητές που υπάρχουν, κοιτάμε να δούμε ποιες έχει ορίσει ο χρήστης. Για αυτές που έχει επιλέξει ο χρήστης εκτυπώνουμε το όνομα της μεταβλητές με μεγάλα γράμματα και επιλέγουμε τους αισθητήρες που ανήκουν σε αυτή.

```

while (@row=$sth->fetchrow_array)
{
    $msg=($language eq "en"?$row[1] $row[3] ($row[4])"
    :"$row[2] $row[3] ($row[4])");
    print $query->checkbox(-name=>"sensor$row[0]",
    -value=>$row[10], -label=>$msg);
    print $query->br;
    $msg=($language eq "gr"?$row[1] $row[3] ($row[4])"
    :"$row[2] $row[3] ($row[4])");
    print $query->b($msg);
    print "&nbsp;$row[5] - $row[6]";
    print $query->br;
}

```

Για τον κάθε αισθητήρα που έχει η βάση, βάζουμε με ένα checkbox το όνομά του, την φίρμα και το μοντέλο του. Στην επόμενη γραμμή προσθέτουμε την περίοδο λειτουργίας του.

```
$sql="SELECT to_char(start_date,'$DATEFORMAT'),".
to_char(end_date,'$DATEFORMAT') FROM sensor_notavail ".
"WHERE sensor_id=$row[0] and variable_id=$variable";
$notavailh=$dbh->prepare($sql)
or syserr("Unable to prepare sensor notavail query:<BR>".
"Database error: $DBI::errstr",1);
$notavailh->execute
or syserr("Could not execute sensor notavail query:<BR>".
"Database Error: $DBI::errstr",1);
$msg=($language eq "gr"?"Περίοδοι ελλείψεων ":"Deficiency period:");
print $query->b($msg);
while (@notavailrow=$notavailh->fetchrow_array)
{
print $query->br("$notavailrow[0] - $notavailrow[1]");
}
print $query->br unless (defined @notavailrow);
syserr("Data sensor_notavail fetching terminated early by error:<BR>".
"Database error: $DBI::errstr",0) if ($DBI::err);
```

Για τις περιόδους ελλείψεων εκτελούμε μία επιπλέον ερώτηση στη βάση για να βρούμε τις περιόδους αυτές για τον συγκεκριμένο αισθητήρα και τις εκτυπώνουμε.

```
$msg=($language eq "gr"?"Ακρίβεια ":"Accuracy:");
print $query->b($msg);
print "$row[7]";
print $query->br;
$msg=($language eq "gr"?"Παρατηρήσεις ":"Remarks:");
print $query->b($msg);
$msg=($language eq "en"?$row[8]:$row[9]);
print $msg;
print $query->br;
print $query->br;
}
print "</BLOCKQUOTE>\n";
}
```

Συνεχίζουμε με την εκτύπωση της ακρίβειας που έχει ο αισθητήρας και τις παρατηρήσεις.

```
print $query->br;
$msg=($language eq "gr"?"Προχωρήστε στη μορφή αρχείου":
"Continue to file format");
print $query->submit(-name=>"proceed", -value=>$msg);
print "&nbsp;";
$msg=($language eq "gr"?"Καθαρισμός φόρμας ":"Clear form");
print $query->reset($msg);

$sth->finish
or syserr("Unable to finish cursor:<BR>Database error: $DBI::errstr", 0);

print $query->endform;
next;
}
```

Στο τέλος της φόρμας, βάζουμε και ένα κουμπί τύπου submit για να προχωρήσει στο επόμενο βήμα και ένα κουμπί για καθαρισμό των επιλογών της φόρμας. Κλείνουμε και τον χειριστή των εντολών, την φόρμα και συνεχίζουμε στην επόμενη γραμμή.

```
print $_;
}
```

```

close(HTMLFH)
or syserr("Unable to close html file $htmlfile:<BR>System error: $!",1);

$dbh->disconnect
or syserr("Disconnect error:<BR>Database error: $DBI::errstr()", 1);

exit 0;

```

Αν οι γραμμές δεν έχουν include ή formdata, τότε εκτυπώνονται ως έχουν. Στο τέλος κλείνεται το αρχείο HTML και γίνεται αποσύνδεση από τη βάση.

4.2.4.5 Τρίτο βήμα ιστορικών δεδομένων histdata3.cgi

Στο τρίτο βήμα των ιστορικών δεδομένων, δίνεται η δυνατότητα στον χρήστη να καθορίσει την μορφή που θα έχει το αρχείο δεδομένων που θα παραλάβει.

```

#!/usr/bin/perl

use CGI;
use DBI;

```

Δήλωση της διαδρομής που βρίσκεται ο διερμηνέας της Perl και χρήση των δομοστοιχείων CGI και DBI.

```

my ($query, $language, $htmlfile, $menuitem, $msg, @sensor_array, $timestep);
my $FORMDATA="<!--FORMDATA-->";
my $INCLUDE="#include";
my $WEB_DIR="/home/httpd/html/newmeteo";
my ($dbservice,$data_source,$dblogin,$dbpassword);
my $configfile="/etc/aeolus/cgi.conf";
my %dbattr=(PrintError=>0, RaiseError=>0, AutoCommit=>0);
my ($dbh, $sth, $sql, @row, @list, %label,$fromdate,$todate, $sensor, $i);
my (@series_array, %series_hash, @parameter, $item);
my $DATEFORMAT="YYYY/MM/DD HH24:MI";
my $type=2;

```

Το \$query χρησιμοποιείται για το στιγμιότυπο του δομοστοιχείου CGI, το \$language για την γλώσσα, το \$htmlfile για τον HTML αρχείο σκελετού της σελίδας, το \$menuitem για το τρέχον όνομα βήματος, το \$msg για την εκτύπωση μηνυμάτων στον χρήστη, το @sensor_array για την αποθήκευση των αναγνωριστικών των αισθητήρων και το \$timestep για το χρονικό βήμα που έχει επιλεγεί. Τα \$FORMDATA και \$INCLUDE είναι συμβολοσειρές που όταν εμφανιστούν στο αρχείο HTML, εκτελούμε συγκεκριμένες λειτουργίες. Τα \$dbservice, \$data_source, \$dblogin και \$dbpassword χρησιμοποιούνται για την σύνδεση στη βάση. Η πλήρης διαδρομή του αρχείου διαμόρφωσης φυλάγεται στο \$configfile και το %dbattr ρυθμίζει διάφορα χαρακτηριστικά για την συμπεριφοράς της βάσης. Το \$dbh είναι ο χειριστής της σύνδεσης στη βάση, το \$sth είναι ο χειριστής των εντολών, το \$sql είναι η συμβολοσειρά με την προς εκτέλεση εντολή, το @row αποθηκεύει την επόμενη γραμμή αποτελεσμάτων, το @list κρατάει τα αναγνωριστικά των χρονικών βημάτων και το %label τις ετικέτες αυτών, το \$fromdate και \$todate κρατούν την ελάχιστη και μέγιστη ημερομηνία που έδωσε ο χρήστης αντίστοιχα, το \$sensor είναι το αναγνωριστικό του αισθητήρα και το \$i είναι προσωρινή μεταβλητή που χρησιμοποιείται στους βρόχους. Το %series_array έχει όλα τα αναγνωριστικά των χρονοσειρών, το %series_hash κρατάει την περιγραφή των αισθητήρων στις δύο γλώσσες, το @parameter παίρνει όλες τις παραμέτρους που υπάρχουν και το \$item είναι βοηθητική για τον αριθμό των αισθητήρων που έχουν επιλεγεί. Η μορφοποίηση της ημερομηνίας γίνεται από το \$DATEFORMAT και το ο τύπος των χρονοσειρών είναι 2.

```

sub syserr
{
  if (defined $_[1])
  {
    unless ($_[1]==0)
    {
      print "ERROR($_[1]): $_[0]\n";
      exit $_[1];
    }
    else
    {
      print "WARNING: $_[0]\n";
    }
  }
  else
  {
    print "Usage: syserr(<message string>, <exit status>)<br>\n";
    print "Exit status:<br>\n";
    print "0\t\tprints a warning and continues script execution<br>\n";
    print "n\t\tprints the error message and exits with the nonzero n status\n";
    exit -1;
  }
}

```

Ρουτίνα χειρισμού λαθών.

```

sub ReadConfig
{
  my $SERVICE="DBService";
  my $LOGIN="DBLogin";
  my $PASSWORD="DBPassword";
  my $keyword;
  my $confforceinsert;

  if (-e $configfile)
  {
    if (-r $configfile)
    {
      open(CFH,$configfile)
        or syserr("Couldn't read from $configfile:\nSystem error: $!",2);

      while (<CFH>)
      {
        if ($_ =~ /^#/)
        {
          next;
        }
        elsif ($_ =~ /\w/)
        {
          if ($_ =~ /$SERVICE/i)
          {
            ($keyword, $dbservice)=split(/=/, $_);
            chomp $dbservice;
            $dbservice=~tr/"'"/d;
          }
          elsif ($_ =~ /$LOGIN/i)
          {
            ($keyword, $dblogin)=split /=/, $_;
            chomp $dblogin;
            $dblogin=~tr/"'"/d;
          }
          elsif ($_ =~ /$PASSWORD/i)
          {

```

```

        ($keyword, $dbpassword)=split /=/, $_;
        chomp $dbpassword;
        $dbpassword=~tr/"'//d;
    }
    else
    {
        syserr("Unsupported configuration line:\n$_",2);
    }
}
}
else
{
    syserr("Config file $configfile exists, but it's unreadable", 2);
}
}
else
{
    syserr("Configuration file $configfile does not exist", 2);
}
close(CFH);
$data_source="DBI:Oracle:$dbservice";
}

```

Ρουτίνα ανάγνωσης αρχείου διαμόρφωσης.

```
$query=new CGI;
```

```
print $query->header;
```

Δημιουργία νέου στιγμιότυπου του αντικειμένου CGI και εκτυπώνονται οι κατάλληλες επικεφαλίδες.

```

$language=$query->param("language");
$timestep=$query->param("timestep");
$fromdate=$query->param("fromdate");
$todate=$query->param("todate");
$htmlfile="$WEB_DIR/histdata/skeleton-$language.shtml";

```

```
@parameter=$query->param;
```

```

$i=0;
while (defined $parameter[$i])
{
    if ($parameter[$i]=~/sensor/i)
    {
        push @series_array, $query->param($parameter[$i]);
    }
    $i++;
}

```

```

$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";
$ENV{NLS_LANG}="AMERICAN_AMERICA.EL8ISO8859P7";
$ENV{ORA_NLS33}="$ENV{ORACLE_HOME}/ocommon/nls/admin/data";

```

Παίρνουμε την γλώσσα, το χρονικό βήμα, την ελάχιστη και μέγιστη ημερομηνία από τις παραμέτρους που έχουν περαστεί στο script. Ρυθμίζεται και το αρχείο HTML. Οι αισθητήρες που έχουν επιλεγεί, τους αποθηκεύουμε με ένα while loop. Τέλος ρυθμίζουμε και τις μεταβλητές περιβάλλοντος.

```
ReadConfig;
```

```
$dbh=DBI->connect($data_source, "$dblogin", "$dbpassword", \%dbattr)
or syserr("Connecting to database failed:<BR>Database Error: $DBI::errstr",1);

$menuitem=($language eq "gr"? "morfh_arxeioy": "file_format");
```

```
open(HTMLFH, $htmlfile)
or syserr("Unable to open html file $htmlfile:<BR>System error: $!",1);
```

```
while (<HTMLFH>)
{
```

Διαβάζουμε τις παραμέτρους σύνδεσης στη βάση από το αρχείο διαμόρφωσης, κάνουμε την σύνδεση, αρχικοποιούμε το τρέχον βήμα που είμαστε και ανοίγουμε το αρχείο HTML για ανάγνωση.

```
if ($_ =~ /$INCLUDE/i)
{
    $_ =~ s/\s*<!--\s*#include\s+virtual=//i;
    $_ =~ s/\s*-->\s*//i;
    $_ =~ tr/' ' //d;
```

Και εδώ αν βρεθεί include θα πρέπει να διερευνήσουμε τις δύο περιπτώσεις, αν πρόκειται δηλαδή για απλό αρχείο που θα συμπεριληφθεί ή για το CGI script που τροποποιεί το μενού.

```
if ($_ =~ /cgi-bin/)
{
    open(MENUFH, "$WEB_DIR"./histdata/steps-$language.html")
    or syserr("Unable to open menu file $_:<BR> System error: $!",2);

    while (<MENUFH>)
    {
        if ($_ =~ /$menuitem/i)
        {
            $_ =~ s#src="/buttons/histdata-
$language/${menuitem}1.jpg"#src="/buttons/histdata-
$language/${menuitem}2.jpg"#i;
        }
        print $_;
    }
    close(MENUFH)
    or syserr("Unable to close steps file $_: $!\n",1);
}
else
{
    open(INCFH, "$WEB_DIR"./$_")
    or syserr("Unable to open include file $WEB_DIR/$_:<BR>System error:
$!",2);

    while (<INCFH>)
    {
        print $_;
    }
    close(INCFH)
    or syserr("Unable to close file $WEB_DIR/$_:<BR>System error: $!",2);
}
next;
}
```

Αν πρόκειται για CGI, τότε διαβάζουμε το αρχείο που έχει το μενού και αλλάζουμε την εικόνα που είναι το τρέχον βήμα με την άλλη που φαίνεται σαν να είναι επιλεγμένη. Σε κάθε άλλη περίπτωση, διαβάζουμε το αρχείο και το συμπεριλαμβάνουμε στο συγκεκριμένο σημείο.


```

or syserr("Could not execute time series query:<BR>Database error: ".
"$DBI::errstr:<BR>\nQuery: $sql", 1);

while (@row=$sth->fetchrow_array)
{
    $series_hash{$series_array[$i]}{"en"}="$row[0] $row[2] ($row[3])";
    $series_hash{$series_array[$i]}{"gr"}="$row[1] $row[2] ($row[3])";
}
}

for $i (0..$#series_array)
{
    push @list, $series_array[$i];
    $label{$series_array[$i]}=$series_hash{$series_array[$i]}{$language};
}

$msg=($language eq "gr"? "Στήλη 1: ":"Column 1: ");
$msg=$query->b($msg);
print $msg;
$msg=($language eq "gr"? "Ημερομηνία και ώρα":"Date and time");
print $msg;
print $query->br;

$i=2;
for $item (keys %series_hash)
{
    $msg=($language eq "gr"? "Στήλη $i: ":"Column $i: ");
    $msg=$query->b($msg);
    print $msg;
    print $query->popup_menu(-name=>"column$i", -values=>\@list,
        -default=>"$list[$i-2]", -labels=>\%label);
    print $query->br;
    $i++;
}

```

Ρυθμίζονται τα ονόματα των αισθητήρων που έχουν επιλεγεί και παρουσιάζονται όλοι μαζί κατά στήλη σε pop up menu, για να μπορεί ο χρήστης να τους αλλάξει την σειρά. Την πρώτη στήλη που έχει την ημερομηνία και την ώρα, ο χρήστης δεν μπορεί να την αλλάξει.

```

print $query->br;
$msg=($language eq "gr"? "Συμπίεση":"Compression");
$msg=$query->b($msg);
print $query->h3($msg);

print $query->p;
@list=();
%label=();
for $i (0..2)
{
    $list[$i]=$i;
}
$label{$list[0]}=($language eq "gr"? "Χωρίς συμπίεση":"No compression");
$label{$list[1]}=($language eq "gr"? "Με συμπίεση ZIP (κατάληξη .zip)"
:"With ZIP compression (.zip suffix)");
$label{$list[2]}=($language eq "gr"? "Με συμπίεση GZIP (κατάληξη .gz)"
:"With GZIP compression (.gz suffix)");
print $query->radio_group(-name=>"compress", -values=>\@list,
    -default=>1, -linebreak=>"true", -labels=>\%label);

```

Δίνεται και η επιλογή στον χρήστη να πάρει το αρχείο και με συμπίεση ZIP ή GZIP.

```

print $query->p;
$msg=($language eq "gr"? "Προχωρήστε στην άδεια χορήγησης":

```

```

"Continue to user licence");
print $query->submit(-name=>"proceed", -value=>$msg);
print "&nbsp;";
$msg=(($language eq "gr"?"Καθαρισμός φόρμας":"Clear form");
print $query->reset($msg);

$sth->finish
or syserr("Unable to finish cursor:<BR>Database error: $DBI::errstr", 0);

print $query->endform;
next;
}

```

Στο τέλος δίνονται και τα δύο κουμπιά, ένα για να προχωρήσει στο επόμενο βήμα και ένα για να καθαρίσει την φόρμα. Κλείνουμε τον χειριστή εντολών και την φόρμα.

```

print $_;
}

close(HTMLFH)
or syserr("Unable to close html file $htmlfile:<BR>System error: $!",1);

$dbh->disconnect
or syserr("Disconnect error:<BR>Database error: $DBI::errstr()", 1);

exit 0;

```

Οι υπόλοιπες γραμμές του αρχείου εκτυπώνονται όπως είναι και στο τέλος κλείνουμε το αρχείο και την σύνδεση με την βάση.

4.2.4.6 Τέταρτο βήμα ιστορικών δεδομένων histdata4.cgi

Σε αυτό το βήμα, παρουσιάζονται όλες οι ρυθμίσεις που έχει κάνει μέχρι στιγμή ο χρήστης και του ζητούνται να προμηθεύσει τα προσωπικά του στοιχεία και να διαβάσει την άδεια χορήγησης.

```

#!/usr/bin/perl

use CGI;
use DBI;

```

Η πρώτη εντολή είναι για τον διερμηνέα της γλώσσας και οι επόμενες δύο για την χρήση των αντίστοιχων δομοστοιχείων.

```

my ($query, $language, $htmlfile, $menuitem, $msg, @sensor_array, $timestep);
my $FORMDATA="<!--FORMDATA-->";
my $INCLUDE="#include";
my $WEB_DIR="/home/httpd/html/newmeteo";
my $TMP_DIR="/tmp/meteo";
my $WEB_DATA_DIR="/histdata/data";
my ($filefromdate,$filetodate,$seperator);
my ($dbservice,$data_source,$dblogin,$dbpassword);
my $configfile="/etc/aeolus/cgi.conf";
my %dbattr=(PrintError=>0, RaiseError=>0, AutoCommit=>0);
my ($dbh, $sth, $sql, @row,$fromdate,$todate, $sensor, $i);
my (@series_array, %series_hash, @parameter, $item);
my ($dateformat,$totalcolumns,$compress,$select,$from,$where);
my ($compress_array,$licencefile);
my $FILEDATEFORMAT="YYYYMMDDHH24MI";
my $FORMDATEFORMAT="HH24:MI DD/MM/YYYY";

```

Το `$query` είναι το στιγμιότυπο του CGI, το `$language` είναι η γλώσσα που χρησιμοποιείται, το `$menuitem` είναι το τρέχον βήμα, το `$msg` είναι μήνυμα για εκτύπωση στον χρήστη, το `$sensor_array` είναι ο πίνακας με τα αναγνωριστικά των αισθητήρων και το `$timestep` είναι το χρονικό βήμα. Τα `$FORMDATA` και `$INCLUDE` είναι συμβολοσειρές που αν βρεθούν στο αρχείο HTML, θα πρέπει να αντιμετωπιστούν διαφορετικά και τα `$WEB_DIR`, `$TMP_DIR` και `$WEB_DATA_DIR` δηλώνουν τα μονοπάτια από καταλόγους του συστήματος που θα χρησιμοποιηθούν. Το `$filefromdate` και το `$filetodate` χρησιμοποιούνται για την επιλογή της ημερομηνίας σύμφωνα με την μορφοποίηση που έχει δώσει ο χρήστης και το `$separator` είναι ο διαχωριστής στηλών που έχει επιλέξει. Τα `$dbservice`, `$data_source`, `$dblogin` και `$dbpassword` χρησιμοποιούνται για την σύνδεση στη βάση. Το `$configfile` για το μονοπάτι του αρχείου διαμόρφωσης και το `%dbattr` για τον καθορισμό των παραμέτρων χρήσης της βάσης. Το `$dbh` είναι ο χειριστής της σύνδεσης στη βάση, το `$sth` ο χειριστής των εντολών, το `$sql` είναι για την εντολή που θα εκτελεστεί, το `@row` είναι για τα αποτελέσματα της εκτέλεσης μίας εντολής, το `$fromdate` και `$todate` είναι η μορφοποίηση του διαστήματος της ημερομηνίας, το `$sensor` είναι το αναγνωριστικό του αισθητήρα και το `$i` είναι προσωρινή μεταβλητή για τους βρόχους. Το `@series_array` κρατάει όλα τα αναγνωριστικά των αισθητήρων που έχει δηλώσει ο χρήστης, το `%series_hash` τα ονόματα αυτών, το `@parameter` χρησιμοποιείται για τις παραμέτρους και το `$item` βοηθητικά για την παρουσίαση των στηλών. Το `$dateformat` είναι η μορφοποίηση της ημερομηνίας που έχει ζητήσει ο χρήστης, το `$totalcolumns` είναι το σύνολο των στηλών, το `$compress` κρατάει την επιλογή για συμπίεση που δόθηκε και τα `$select`, `$from` και `$where` είναι βοηθητικές μεταβλητές για την κατασκευή μίας πολύπλοκης εντολής SQL. Το `$compress_array` έχει τις επιλογές για συμπίεση, το `$licencefile` έχει το μονοπάτι από το αρχείο με την άδεια χορήγησης και οι δύο μορφοποιήσεις της ημερομηνίας του αρχείου και της φόρμας περιγράφονται από τις μεταβλητές `$FILEDATEFORMAT` και `$FORMDATEFORMAT`.

```
sub syserr
{
    if (defined $_[1])
    {
        unless ($_[1]==0)
        {
            print "ERROR($_[1]): $_[0]\n";
            exit $_[1];
        }
        else
        {
            print "WARNING: $_[0]\n";
        }
    }
    else
    {
        print "Usage: syserr(<message string>, <exit status>)\n";
        print "Exit status:\n";
        print "0\t\tprints a warning and continues script execution\n";
        print "n\t\tprints the error message and exits with the nonzero n status\n";
        exit -1;
    }
}
```

Ρουτίνα χειρισμού λαθών.

```
sub ReadConfig
{
    my $SERVICE="DBService";
    my $LOGIN="DBLogin";
    my $PASSWORD="DBPassword";
    my $keyword;
    my $confforceinsert;
```

```

if (-e $configfile)
{
  if (-r $configfile)
  {
    open(CFH,$configfile)
      or syserr("Couldn't read from $configfile:\nSystem error: $!",2);

    while (<CFH>)
    {
      if ($_ =~ /^#/ )
      {
        next;
      }
      elsif ($_ =~ /\w/ )
      {
        if ($_ =~ /$SERVICE/i)
        {
          ($keyword, $dbservice)=split(/=/, $_);
          chomp $dbservice;
          $dbservice=~tr/"'//d;
        }
        elsif ($_ =~ /$LOGIN/i)
        {
          ($keyword, $dblogin)=split /=/, $_;
          chomp $dblogin;
          $dblogin=~tr/"'//d;
        }
        elsif ($_ =~ /$PASSWORD/i)
        {
          ($keyword, $dbpassword)=split /=/, $_;
          chomp $dbpassword;
          $dbpassword=~tr/"'//d;
        }
        else
        {
          syserr("Unsupported configuration line:\n$_",2);
        }
      }
    }
  }
  else
  {
    syserr("Config file $configfile exists, but it's unreadable", 2);
  }
}
else
{
  syserr("Configuration file $configfile does not exist", 2);
}
close(CFH);
$data_source="DBI:Oracle:$dbservice";
}

```

Ρουτίνα ανάγνωσης αρχείου διαμόρφωσης.

```

$query=new CGI;

print $query->header;

$language=$query->param("language");
$timestep=$query->param("timestep");
$fromdate=$query->param("fromdate");

```

```
$todate=$query->param("todate");
$dateformat=$query->param("dateformat");
$separator=$query->param("separator");
$compress=$query->param("compress");
$htmlfile="$WEB_DIR/histdata/skeleton-$language.shtml";
```

Δημιουργία του στιγμιότυπου \$query και εκτύπωση των επικεφαλίδων. Οι παράμετροι που έχουν περαστεί στο CGI script αποθηκεύονται σε μεταβλητές στο πρόγραμμα και ρυθμίζεται το αρχείο HTML.

```
@parameter=$query->param;
$i=0;
while (defined $parameter[$i])
{
  if ($parameter[$i]=~/column/i)
  {
    push @series_array, $query->param($parameter[$i]);
  }
  $i++;
}

$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";
$ENV{NLS_LANG}="AMERICAN_AMERICA.EL8ISO8859P7";
$ENV{ORA_NLS33}="$ENV{ORACLE_HOME}/ocommon/nls/admin/data";
```

```
ReadConfig;
```

```
$dbh=DBI->connect($data_source, "$dblogin", "$dbpassword", \%dbattr)
or syserr("Connecting to database failed:<BR>Database Error: $DBI::errstr",1);
```

```
$menuitem=($language eq "gr"?"adeia_xorhghshs":"user_licence");
```

Οι παράμετροι με τους αισθητήρες αποθηκεύονται στο @series_array και ρυθμίζονται οι μεταβλητές περιβάλλοντος της Oracle. Διαβάζεται το αρχείο διαμόρφωσης, γίνεται σύνδεση στη βάση και αρχικοποιείται το \$menuitem στο τρέχον βήμα.

```
$sql="SELECT to_char(data_date,'$FILEDATEFORMAT') FROM data_proc ".
"WHERE data_date=to_date('$fromdate','$FORMDATEFORMAT')";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare date statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute date statement:<BR>Database error: $DBI::errstr",
1);
@row=$sth->fetchrow_array;
$filefromdate=$row[0];
```

```
$sql="SELECT to_char(data_date,'$FILEDATEFORMAT') FROM data_proc ".
"WHERE data_date=to_date('$todate','$FORMDATEFORMAT')";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare date statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute date statement:<BR>Database error: $DBI::errstr",
1);
@row=$sth->fetchrow_array;
$filetodate=$row[0];
```

Παίρνουμε την μέγιστη και την ελάχιστη ημερομηνία που ζήτησε ο χρήστης από την βάση, μετατρέποντας την μορφοποίηση σε αυτή που δήλωσε στο προηγούμενο βήμα.

```

$sql="SELECT s.name,s.gr_name FROM time_series t,time_step s ".
"WHERE series_id=$series_array[0] and t.step_id=s.step_id";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare step statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute step statement:<BR>Database error: $DBI::errstr",
1);
@row=$sth->fetchrow_array;
$timestep=($language eq "en"?$row[0]:$row[1]);
$totalcolumns=$#series_array+2;

open(HTMLFH,$htmlfile)
or syserr("Unable to open html file $htmlfile:<BR>System error: $!",1);

while (<HTMLFH>)
{

```

Παίρνουμε και το όνομα του χρονικού βήματος από την βάση. Ανοίγουμε το αρχείο με τον σκελετό της σελίδας σε HTML και αρχίζουμε να ελέγχουμε τα σημεία που θέλουν επεξεργασία.

```

if ($_~/ $INCLUDE/i)
{
    $_~/s/\s*<!--\s*#include\s+virtual=//i;
    $_~/s/\s*-->\s*//i;
    $_~/tr/' ' //d;
    if ($_~/cgi-bin/)
    {
        open(MENUFH, "$WEB_DIR"./histdata/steps-$language.html")
        or syserr("Unable to open menu file $_:<BR> System error: $!",2);

        while (<MENUFH>)
        {
            if ($_~/ $menuitem/i)
            {
                $_~/s#src="/buttons/histdata-
                $language/{menuitem}1.jpg"#src="/buttons/histdata-
                $language/{menuitem}2.jpg"#i;
            }
            print $_;
        }
        close(MENUFH)
        or syserr("Unable to close steps file $_: $!\n",1);
    }
    else
    {
        open(INCFH, "$WEB_DIR"./$_)
        or syserr("Unable to open include file $WEB_DIR/$_<BR>System error:
        $!",2);

        while (<INCFH>)
        {
            print $_;
        }
        close(INCFH)
        or syserr("Unable to close file $WEB_DIR/$_:<BR>System error: $!",2);
    }
    next;
}
}

```

Αν βρούμε include, τότε το συμπεριλαμβάνουμε στο συγκεκριμένο σημείο, εκτός και αν είναι CGI που θα πρέπει να αλλάξουμε τις εικόνες του μενού.

```

elseif ($_~/ $FORMDATA/i)
{
    print $query->startform(-method=>POST, -action=>"/cgi-
bin/newmeteo/histdata5.cgi");
    print $query->hidden(-name=>"language", -default=>$language);
    print $query->hidden(-name=>"timestep", -default=>$timestep);
    print $query->hidden(-name=>"fromdate", -default=>$filefromdate);
    print $query->hidden(-name=>"todate", -default=>$filefromdate);
    print $query->hidden(-name=>"dateformat", -default=>$dateformat);
    print $query->hidden(-name=>"seperator", -default=>$seperator);
    print $query->hidden(-name=>"compress", -default=>$compress);
    foreach $i (0..$#series_array)
    {
        $item=$i+2;
        print $query->hidden(-name=>"column$item", -default=>$series_array[$i]);
    }

    $msg=($language eq "gr"? "4. Άδεια χορήγησης": "4. User licence");
    $msg=$query->b($msg);
    print $query->h2($msg);
    print $query->br;
    print "\n";
}

```

Αν είμαστε στο σημείο που πρέπει να εκτυπώσουμε την φόρμα, την ξεκινάμε με την μέθοδο POST που μας πάει στο επόμενο και τελευταίο βήμα. Περνάμε κρυφά όλες τις παραμέτρους που χρειάζεται το τελευταίο βήμα.

```

$msg=($language eq "gr"? "Ζητήσατε να παραλάβετε το αρχείο με τις εξής ".
"πληροφορίες: ":"You have asked to get a file with the following ".
"information: ");
print "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;";
print $msg;
print $query->p;
print "\n";

$msg=($language eq "gr"? "Χρονικό διάστημα: ":"Time duration: ");
print $query->b($msg);
print "$fromdate - $todate";
print $query->br;
print "\n";

$msg=($language eq "gr"? "Χρονικό βήμα: ":"Time step: ");
print $query->b($msg);
print $timestep;
print $query->br;
print "\n";

$msg=($language eq "gr"? "Μορφή ημερομηνίας και ώρας: ":"
Date and time format: ");
print $query->b($msg);
print $dateformat;
print $query->br;
print "\n";

$msg=($language eq "gr"? "Διαχωριστής στηλών: ":"Column seperator: ");
print $query->b($msg);
print $seperator;
print $query->br;
print "\n";

```

Εκτυπώνονται στην αρχή της φόρμας οι πληροφορίες που έχει δώσει μέχρι αυτό το βήμα για το χρονικό διάστημα, το χρονικό βήμα, την μορφή της ημερομηνίας και της ώρας και τον διαχωριστή στηλών.


```

$msg=(($language eq "gr"? "Στήλη 1: ":"Column 1: ");
print $query->b($msg);
$msg=(($language eq "gr"? "Ημερομηνία και ώρα:" "Date and time");
print $msg;
print $query->br;
print "\n";

for $i (0..$#series_array)
{
    $item=$i+2;
    $msg=(($language eq "gr"? "Στήλη $item: ":"Column $item: ");
    print $query->b($msg);

    $sql="SELECT v.name,v.gr_name,s.firm,s.model FROM sensor s, ".
    "variable v, time_series t WHERE t.series_id=$series_array[$i] and ".
    "t.sensor_id=s.sensor_id and t.variable_id=v.variable_id";
    $sth=$dbh->prepare($sql)
    or syserr("Unable to prepare time series query:<BR>Database error: ".
    "$DBI::errstr", 1);
    $sth->execute
    or syserr("Could not execute time series query:<BR>Database error: ".
    "$DBI::errstr:<BR>\nQuery: $sql", 1);

    while (@row=$sth->fetchrow_array)
    {
        $series_hash{$series_array[$i]}{"en"}="$row[0] $row[2] ($row[3])";
        $series_hash{$series_array[$i]}{"gr"}="$row[1] $row[2] ($row[3])";
    }
    syserr("Time series fetching terminated early by error:<BR>Database ".
    "error: $DBI::errstr",0) if ($DBI::err);

    print $series_hash{$series_array[$i]}{$language};
    print $query->br;
    print "\n";
}

```

Η λίστα των επιλογών του χρήστη συνεχίζει με την σειρά των στηλών που θα έχουν οι αισθητήρες.

```

$msg=(($language eq "gr"? "Συμπίεση: ":"Compression: ");
print $query->b($msg);
$compress_array[0]=($language eq "gr"? "Χωρίς συμπίεση:" "No compression");
$compress_array[1]=($language eq "gr"? "Με συμπίεση ZIP"
:"With ZIP compression");
$compress_array[2]=($language eq "gr"? "Με συμπίεση GZIP"
:"With GZIP compression");
print "$compress_array[$compress]";
print $query->br;
print "\n";

```

Και τελειώνει με την συμπίεση του αρχείου που ζήτησε.

```

$msg=(($language eq "gr"? "Προσωπικές πληροφορίες:"
:"Personal information:\n");
$msg=$query->b($msg);
print $query->h3($msg);
print "\n";

```

```

$msg=(($language eq "gr"? "Είναι υποχρεωτική η συμπλήρωση όλων των πεδίων. ".
"Η ηλεκτρονική σας διεύθυνση είναι αναγκαία για να σας αποσταλεί το URL με".
"τα δεδομένα. Δεν πρόκειται στο μέλλον να σας στείλουμε τίποτε άλλο σ' ".
"αυτή τη διεύθυνση, εκτός αν προκύψει κάτι (π.χ. αν ανακαλυφθεί λάθος) ".
"σχετικό με τα δεδομένα που θα σας έχουν αποσταλεί."

```

```

:"All fields must be filled in. Your e-mail address is necessary so that".
"the URL of the data are sent. No other message will be sent in the ".
"future to this address, unless something comes out (e.g. an error is ".
"discovered) concerning the data you will already have received.");
print $msg;
print $query->p;
print "\n";

```

Στην συνέχεια ζητάμε από τον χρήστη να δώσει τα προσωπικά του στοιχεία.

```

print $query->table;
print "<TR>";
$msg=($language eq "gr"?"Όνομα: ":"Name: ");
print $query->td($query->i($query->b($msg)));
print $query->td($query->textfield(-name=>"name", -size=>30,
-maxlength=>40, -override=>1));
print "</TR>\n";

print "<TR>";
$msg=($language eq "gr"?"Επώνυμο: ":"Surname: ");
print $query->td($query->i($query->b($msg)));
print $query->td($query->textfield(-name=>"surname", -size=>30,
-maxlength=>40, -override=>1));
print "</TR>\n";

print "<TR>";
$msg=($language eq "gr"?"Ιδιότητα: ":"Occupation: ");
print $query->td($query->i($query->b($msg)));
print $query->td($query->textfield(-name=>"occupation", -size=>30,
-maxlength=>40, -override=>1));
print "</TR>\n";

print "<TR>";
$msg=($language eq "gr"?"Εταιρία: ":"Company: ");
print $query->td($query->i($query->b($msg)));
print $query->td($query->textfield(-name=>"company", -size=>30,
-maxlength=>40, -override=>1));
print "</TR>\n";

print "<TR>";
$msg=($language eq "gr"?"Διεύθυνση e-mail: ":"E-mail address: ");
print $query->td($query->i($query->b($msg)));
print $query->td($query->textfield(-name=>"email", -size=>30,
-maxlength=>70, -override=>1));
print "</TR>\n";

print "<TR>";
$msg=($language eq "gr"?"Σκοπός: ":"Purpose: ");
print '<TD VALIGN="top">';
print $query->i($query->b($msg));
print "</TD>";
print $query->td($query->textarea(-name=>"purpose", -rows=>10,
-columns=>30, -override=>1));
print "</TR>\n";

print "</TABLE>\n";

```

Στα πεδία που έχει να συμπληρώσει και είναι υποχρεωτικά, είναι το όνομά του, το επίθετό του, την ιδιότητα που έχει, την εταιρία που βρίσκεται, την διεύθυνση e-mail και τον σκοπό που θέλει τα δεδομένα.

```

$licencefile=$WEB_DIR."/histdata/licence-$language.html";
open (LCNFH,$licencefile) or

```

```

syserr("Unable to open licence file $licencefile:<BR>System error: $!",1);
while (<LCNFH>)
{
    print $_;
}
close(LCNFH) or
syserr("Unable to close licence file $licencefile:<BR>System error: $!",1);

$msg=($language eq "gr"?"Το σύστημα θα επεξεργαστεί την αίτησή σας και ".
"θα σας αποστείλει e-mail με τις πληροφορίες της δομής του αρχείου και ".
"το URL, όπου θα μπορέσετε να παραλάβετε τα δεδομένα."
:"The data URL will be automatically e-mailed when you click the ".
"following button. Your clicking the button implies your acceptance ".
"of the above terms and conditions.");
print $query->p($query->strong($msg));

```

Ακολουθεί η άδεια χορήγησης και ένα σύντομο μήνυμα που να ειδοποιεί τον χρήστη για το επόμενο βήμα.

```

print $query->p;
$msg=($language eq "gr"?"Υποβολή αίτησης":"Submit application");
$msg=$query->submit(-name=>"proceed", -value=>$msg);
print $query->center($msg);

$sth->finish
or syserr("Unable to finish cursor:<BR>Database error: $DBI::errstr", 0);

print $query->endform;

next;
}

```

Εκτυπώνεται ένα κουμπί για να προχωρήσει και κλείνουμε την φόρμα και τον χειριστή εντολών.

```

print $_;
}

close(HTMLFH)
or syserr("Unable to close html file $htmlfile:<BR>System error: $!",1);

$dbh->disconnect
or syserr("Disconnect error:<BR>Database error: $DBI::errstr()", 1);

exit 0;

```

Οι υπόλοιπες γραμμές του αρχείου εκτυπώνονται όπως είναι και στην συνέχεια το κλείνουμε και γίνεται αποσύνδεση από την βάση.

4.2.4.7 Πέμπτο βήμα ιστορικών δεδομένων histdata5.cgi

Στο τελευταίο βήμα ξεκινάει η επεξεργασία των στοιχείων που έδωσε ο χρήστης και του εκτυπώνεται ένα μήνυμα, ότι θα του αποσταλεί ηλεκτρονικό γράμμα με την διεύθυνση URL των δεδομένων που ζήτησε.

```

#!/usr/bin/perl

my $VERSION="1.0.7";

use CGI;
use DBI;

```

```
use Mail::Send;
```

Δίνεται η το μονοπάτι του διερμηνέα της Perl, η έκδοση του συστήματος που στέλνει mail και γίνεται χρήση των δομοστοιχείων CGI, DBI και Mail::Send.

```
my ($query, $language, $htmlfile, $menuitem, $msg, @sensor_array, $timestep);
my $FORMDATA="<!--FORMDATA-->";
my $INCLUDE="#include";
my $WEB_DIR="/home/httpd/html/newmeteo";
my $TMP_DIR="/tmp/meteo";
my $WEB_DATA_DIR="/histdata/data";
my $ZIP="/usr/bin/zip";
my $GZIP="/bin/gzip";
my ($datafile, $filefromdate, $filetodate, $seperator);
my ($dbservice, $data_source, $dblogin, $dbpassword);
my $configfile="/etc/aeolus/cgi.conf";
my %dbattr=(PrintError=>0, RaiseError=>0, AutoCommit=>0);
my ($dbh, $sth, $sql, @row, $fromdate, $todate, $sensor, $i);
my (@series_array, %series_hash, @parameter, $item);
my ($dateformat, $totalcolumns, $compress, $select, $from, $where);
my ($compress_array, $licencefile, $mfh, $mail, $subject);
my $FILEDATEFORMAT="YYYYMMDDHH24MI";
my $FORMDATEFORMAT="HH24:MI DD/MM/YYYY";
```

Το \$query είναι το στιγμιότυπο του CGI, το \$language είναι η γλώσσα που χρησιμοποιείται, το \$menuitem είναι το τρέχον βήμα, το \$msg είναι μήνυμα για εκτύπωση στον χρήστη, το \$sensor_array είναι ο πίνακας με τα αναγνωριστικά των αισθητήρων και το \$timestep είναι το χρονικό βήμα. Τα \$FORMDATA και \$INCLUDE είναι συμβολοσειρές που αν βρεθούν στο αρχείο HTML, θα πρέπει να αντιμετωπιστούν διαφορετικά και τα \$WEB_DIR, \$TMP_DIR και \$WEB_DATA_DIR δηλώνουν τα μονοπάτια από καταλόγους του συστήματος που θα χρησιμοποιηθούν και τα \$ZIP και \$GZIP την διαδρομή των προγραμμάτων που θα γίνει η συμπίεση. Το \$datafile χρησιμοποιείται για την δημιουργία του αρχείου που θα κρατήσει τα δεδομένα, το \$filefromdate και το \$filetodate χρησιμοποιούνται για την επιλογή της ημερομηνίας σύμφωνα με την μορφοποίηση που έχει δώσει ο χρήστης και το \$seperator είναι ο διαχωριστής στηλών που έχει επιλέξει. Τα \$dbservice, \$data_source, \$dblogin και \$dbpassword χρησιμοποιούνται για την σύνδεση στη βάση. Το \$configfile για το μονοπάτι του αρχείου διαμόρφωσης και το %dbattr για τον καθορισμό των παραμέτρων χρήσης της βάσης. Το \$dbh είναι ο χειριστής της σύνδεσης στη βάση, το \$sth ο χειριστής των εντολών, το \$sql είναι για την εντολή που θα εκτελεστεί, το @row είναι για τα αποτελέσματα της εκτέλεσης μίας εντολής, το \$fromdate και \$todate είναι η μορφοποίηση του διαστήματος της ημερομηνίας, το \$sensor είναι το αναγνωριστικό του αισθητήρα και το \$i είναι προσωρινή μεταβλητή για τους βρόχους. Το @series_array κρατάει όλα τα αναγνωριστικά των αισθητήρων που έχει δηλώσει ο χρήστης, το %series_hash τα ονόματα αυτών, το @parameter χρησιμοποιείται για τις παραμέτρους και το \$item βοηθητικά για την παρουσίαση των στηλών. Το \$dateformat είναι η μορφοποίηση της ημερομηνίας που έχει ζητήσει ο χρήστης, το \$totalcolumns είναι το σύνολο των στηλών, το \$compress κρατάει την επιλογή για συμπίεση που δόθηκε και τα \$select, \$from και \$where είναι βοηθητικές μεταβλητές για την κατασκευή μίας πολύπλοκης εντολής SQL. Το \$compress_array έχει τις επιλογές για συμπίεση, το \$licencefile έχει το μονοπάτι από το αρχείο με την άδεια χορήγησης, το \$mfh είναι ο χειριστής του αρχείου του ταχυδρομείου, το \$mail είναι το στιγμιότυπο του δομοστοιχείου Mail::Send και το \$subject είναι το θέμα που έχει το γράμμα. Οι δύο μορφοποιήσεις της ημερομηνίας του αρχείου και της φόρμας περιγράφονται από τις μεταβλητές \$FILEDATEFORMAT και \$FORMDATEFORMAT.

```
my ($surname, $name, $occupation, $company, $email, $purpose, $remotehost);
my $TO='The Meteo Team <meteo@meteo.ntua.gr>';
my $FROM='Online Weather Data <meteo@meteo.ntua.gr>';
my $SUBJECT="METEO: ";
my $XMAILER="Online Weather Mailer $VERSION";
```

```
my $CONTENTTYPE="text/plain; charset=";
my $TRANSFERENCODING="bit";
my $HOST="/usr/bin/host";
```

Το \$surname είναι το επίθετο του χρήστη, το \$name το όνομα, το \$occupation η απασχόλησή του, το \$company η εταιρία του, το \$email η ηλεκτρονική του διεύθυνση, το \$purpose ο σκοπός του και το \$remotehost είναι η IP διεύθυνση που έχει. Το \$TO είναι η επικεφαλίδα που ορίζει που θα σταλεί το γράμμα, το \$FROM από ποιον θα είναι, το \$SUBJECT το θέμα, το \$XMAILER το πρόγραμμα που στέλνει το γράμμα, το \$CONTENTTYPE είναι η μορφή που θα έχει, το \$TRANSFERENCODING είναι η κωδικοποίηση και το \$HOST είναι το μονοπάτι της εντολής host του Unix.

```
sub syserr
{
  if (defined $_[1])
  {
    unless ($_[1]==0)
    {
      print "ERROR($_[1]): $_[0]\n";
      exit $_[1];
    }
    else
    {
      print "WARNING: $_[0]\n";
    }
  }
  else
  {
    print "Usage: syserr(<message string>, <exit status>)\n";
    print "Exit status:\n";
    print "0\t\tprints a warning and continues script execution\n";
    print "n\t\tprints the error message and exits with the nonzero n status\n";
    exit -1;
  }
}
```

Ρουτίνα για τον χειρισμό λαθών.

```
sub ReadConfig
{
  my $SERVICE="DBService";
  my $LOGIN="DBLogin";
  my $PASSWORD="DBPassword";
  my $keyword;
  my $confforceinsert;

  if (-e $configfile)
  {
    if (-r $configfile)
    {
      open(CFH,$configfile)
        or syserr("Couldn't read from $configfile:\nSystem error: $!",2);

      while (<CFH>)
      {
        if ($_ =~ /^#/)
        {
          next;
        }
        elsif ($_ =~ /\w/)
        {
          if ($_ =~ /$SERVICE/i)
```

```

    {
        ($keyword, $dbservice)=split(/=/, $_);
        chomp $dbservice;
        $dbservice=~tr/"'//d;
    }
    elsif ($_=~/$LOGIN/i)
    {
        ($keyword, $dblogin)=split /=/, $_;
        chomp $dblogin;
        $dblogin=~tr/"'//d;
    }
    elsif ($_=~/$PASSWORD/i)
    {
        ($keyword, $dbpassword)=split /=/, $_;
        chomp $dbpassword;
        $dbpassword=~tr/"'//d;
    }
    else
    {
        syserr("Unsupported configuration line:\n$_",2);
    }
}
}
}
else
{
    syserr("Config file $configfile exists, but it's unreadable", 2);
}
}
else
{
    syserr("Configuration file $configfile does not exist", 2);
}
close(CFH);
$data_source="DBI:Oracle:$dbservice";
}

```

Ρουτίνα για την ανάγνωση παραμέτρων σύνδεσης στη βάση από αρχείο διαμόρφωσης.

```

$query=new CGI;

$language=$query->param("language");
$timestep=$query->param("timestep");
$fromdate=$query->param("fromdate");
$today=$query->param("today");
$dateformat=$query->param("dateformat");
$separator=$query->param("separator");
$compress=$query->param("compress");
$name=$query->param("name");
$surname=$query->param("surname");
$occupation=$query->param("occupation");
$company=$query->param("company");
$email=$query->param("email");
$purpose=$query->param("purpose");
$htmlfile="$WEB_DIR/histdata/skeleton-$language.shtml";
@parameter=$query->param;
$i=0;
while (defined $parameter[$i])
{
    if ($parameter[$i]=~/column/i)
    {
        push @series_array, $query->param($parameter[$i]);
    }
}

```

```

    $i++;
}

```

Δημιουργία του στιγμιότυπου \$query και ανάκτηση όλων των παραμέτρων που έχουν περάσει από το προηγούμενο βήμα.

```

$compress_array[0]=($language eq "gr"?"Χωρίς συμπίεση":"No compression");
$compress_array[1]=($language eq "gr"?"Με συμπίεση ZIP"
:"With ZIP compression");
$compress_array[2]=($language eq "gr"?"Με συμπίεση GZIP"
:"With GZIP compression");

```

```

$ENV{ORACLE_HOME}="/home/oracle/ora816";
$ENV{ORACLE_BASE}="/home/oracle";
$ENV{NLS_LANG}="AMERICAN_AMERICA.EL8ISO8859P7";
$ENV{ORA_NLS33}="$ENV{ORACLE_HOME}/ocommon/nls/admin/data";

```

```
ReadConfig;
```

Αρχικοποίηση του πίνακα με τις περιγραφές των συμπίεσεων, καθώς και των μεταβλητών περιβάλλοντος. Στην συνέχεια το πρόγραμμα χωρίζει σε δύο ανεξάρτητες διεργασίες. Η διεργασία παιδί λαμβάνει πρόνοια να μεριμνήσει για την ανάκτηση των δεδομένων και η διεργασία πατέρας συνεχίζει και εκτυπώνει ένα μήνυμα για πληροφόρηση του χρήστη.

```

if (fork==0)
{
    open STDIN, "/dev/null";
    open STDOUT, ">/dev/null";
    $dbh=DBI->connect($data_source, "$dblogin", "$dbpassword", \%dbattr)
    or syserr("Connecting to database failed:<BR>Database Error: $DBI::errstr",1);

```

Η διεργασία παιδί όταν ξεκινάει, αποδεσμεύεται τελείως από τον πατέρα, κλείνοντας την στάνταρ είσοδο και έξοδο του. Εγκαθιστά την σύνδεση με τη βάση.

```

$sql="SELECT to_char(data_date,'$FILEDATEFORMAT') FROM data_proc ".
"WHERE data_date=to_date('$fromdate','$FORMDATEFORMAT')";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare date statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute date statement:<BR>Database error: ".
"$DBI::errstr", 1);
@row=$sth->fetchrow_array;
$filefromdate=$row[0];

$sql="SELECT to_char(data_date,'$FILEDATEFORMAT') FROM data_proc ".
"WHERE data_date=to_date('$todate','$FORMDATEFORMAT')";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare date statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute date statement:<BR>Database error: $DBI::errstr",
1);
@row=$sth->fetchrow_array;
$filetodate=$row[0];

```

Παίρνουμε την ελάχιστη και μέγιστη ημερομηνία, στην μορφή που την θέλει ο χρήστης.

```

$sql="SELECT s.name,s.gr_name FROM time_series t,time_step s ".
"WHERE series_id=$series_array[0] and t.step_id=s.step_id";
$sth=$dbh->prepare($sql)

```

```

or syserr("Unable to prepare step statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute step statement:<BR>Database error: $DBI::errstr",
1);
@row=$sth->fetchrow_array;
$timestep=($language eq "en"?$row[0]:$row[1]);
$totalcolumns=$#series_array+2;

$datafile="ntuameteo-$filefromdate-$filetodate-$row[0]-$totalcolumns";

```

Παίρνουμε από τη βάση το χρονικό βήμα που έχει δηλώσει ο χρήστης και ορίζουμε το όνομα του αρχείου.

```

open(DFH,">$TMP_DIR/$datafile")
or syserr("Unable to open datafile $datafile for writing:<BR>System error:
$!",
1);

$select="to_char(d.data_date,'$dateformat')";
$from="data_proc d";
$where="d.data_date>=to_date('$fromdate','$FORMDATEFORMAT') AND ".
"d.data_date<=to_date('$todate','$FORMDATEFORMAT')";

for $i (0..$#series_array)
{
$select=$select.",t$i.value";
$from=$from.",data_proc t$i";
$where=$where." AND t$i.data_date(+)=d.data_date";
$where=$where." AND t$i.series_id(+)=${series_array[$i]}";
}
$sql="SELECT DISTINCT $select FROM $from WHERE $where";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare data statement:<BR>Database error: $DBI::errstr",
1);
$sth->execute
or syserr("Unable to execute data statement:<BR>Database error: ".
"$DBI::errstr",1);

```

Ανοίγουμε το αρχείο για τα δεδομένα για γράψιμο και προετοιμάζουμε την ερώτηση για ανάκτηση των δεδομένων που ζήτησε ο χρήστης.

```

while (@row=$sth->fetchrow_array)
{
print DFH "$row[0]";
for $i (0..$#series_array)
{
print DFH "$separator$row[$i+1]";
}
print DFH "\n";
}
syserr("Data fetching terminated early by error:<BR>Database error:".
"$DBI::errstr",0) if ($DBI::err);

close(DFH);

```

Τα αποτελέσματα της εκτέλεσης, με ένα while loop τα γράφουμε στο αρχείο και όταν ολοκληρωθεί η διαδικασία το κλείνουμε.

```

if ($compress==0)
{
# do nothing

```



```

}
elsif ($compress==1)
{
  ` $ZIP $TMP_DIR/$datafile $TMP_DIR/$datafile`;
  $datafile=$datafile.".zip";
}
elsif ($compress==2)
{
  ` $GZIP $TMP_DIR/$datafile`;
  $datafile=$datafile.".gz";
}
else
{
  syserr("Unknown compression method to follow",1);
}

```

Ελέγχουμε την συμπίεση που επιθυμεί ο χρήστης και ενεργούμε ανάλογα.

```

$mail=new Mail::Send or
syserr("Unable to create new mail object",1);
$mail->set("From",$FROM);
$msg=($language eq "gr"? "Η αίτησή σας έγινε δεκτή"
:"Your application has been accepted");
$subject=$SUBJECT.$msg;
$mail->subject($subject);
$mail->to("$name $surname <$email>");
$mail->set("X-Mailer",$XMAILER);
$CONTENTTYPE=($language eq "gr"?$CONTENTTYPE."iso-8859-7"
:$CONTENTTYPE."iso-8859-1");
$mail->set("Content-Type",$CONTENTTYPE);
$TRANSFERENCODING=($language eq "gr"? "8".$TRANSFERENCODING
:"7".$TRANSFERENCODING);
$mail->set("Content-Transfer-Encoding",$TRANSFERENCODING);
$mfh=$mail->open('sendmail') or
syserr("Unable to open mail file handle",1);

```

Δημιουργούμε το στιγμιότυπο του δομοστοιχείου Mail::Send \$mail και ρυθμίζουμε τις επικεφαλίδες του μηνύματος.

```

$msg=($language eq "gr"
?"Αγαπητέ κύριε/κυρία $name $surname,\n\n".
"\tΗ επεξεργασία των δεδομένων ολοκληρώθηκε και μπορείτε να τα\n".
"παραλάβετε από την ακόλουθη διεύθυνση:\n\n".
"http://meteo.ntua.gr$WEB_DATA_DIR/$datafile\n\n".
"Το αρχείο που θα παραλάβετε έχει την ακόλουθη μορφή:\n\n"
:"Dear sir/madam $name $surname,\n\n".
"\tData processing has been completed and you may get your data\n".
"from the following address:\n\n".
"http://meteo.ntua.gr$WEB_DATA_DIR/$datafile\n\n".
"Data file format, which you will get, is as follows:\n\n");
print $mfh $msg;
$msg=($language eq "gr"? "Χρονικό διάστημα: $fromdate - $todate\n"
:"Time duration: $fromdate - $todate\n");
print $mfh $msg;
$msg=($language eq "gr"? "Χρονικό βήμα: $timestep\n" : "Time step: $timestep\n");
print $mfh $msg;
$msg=($language eq "gr"? "Μορφή ημερομηνίας και ώρας: $dateformat\n" :
"Date and time format: $dateformat\n");
print $mfh $msg;
$msg=($language eq "gr"? "Διαχωριστής στηλών: $separator\n"
:"Column separator: $separator\n");
print $mfh $msg;
$msg=($language eq "gr"? "Στήλη 1: ":"Column 1: ");

```

```

print $mfh $msg;
$msg=($language eq "gr"?"Ημερομηνία και ώρα\n":"Date and time\n");
print $mfh $msg;
for $i (0..$#series_array)
{
    $item=$i+2;
    $msg=($language eq "gr"?"Στήλη $item: ":"Column $item: ");
    print $mfh $msg;

    $sql="SELECT v.name,v.gr_name,s.firm,s.model FROM sensor s, ".
        "variable v, time_series t WHERE t.series_id=$series_array[$i] and ".
        "t.sensor_id=s.sensor_id and t.variable_id=v.variable_id";
    $sth=$dbh->prepare($sql)
    or syserr("Unable to prepare time series query:<BR>Database error: ".
        "$DBI::errstr", 1);
    $sth->execute
    or syserr("Could not execute time series query:<BR>Database error: ".
        "$DBI::errstr:<BR>\nQuery: $sql", 1);

    while (@row=$sth->fetchrow_array)
    {
        $series_hash{$series_array[$i]}{"en"}="$row[0] $row[2] ($row[3])";
        $series_hash{$series_array[$i]}{"gr"}="$row[1] $row[2] ($row[3])";
    }
    syserr("Time series fetching terminated early by error:<BR>Database ".
        "error: $DBI::errstr",0) if ($DBI::err);

    print $mfh $series_hash{$series_array[$i]}{$language};
    print $mfh "\n";
}
$msg=($language eq "gr"?"Συμπύεση: $compress_array[$compress]\n"
:"Compression: $compress_array[$compress]\n");
print $mfh $msg;
$licencefile=$WEB_DIR."/histdata/licence-$language.html";
open (LCNFH,$licencefile) or
syserr("Unable to open licence file $licencefile:<BR>System error: $!",1);
while (<LCNFH>)
{
    if ($_ =~ /<.*>/)
    {
        $_ =~ s/<LI>/* /g;
        $_ =~ s/<[\wA-Ωα-ωέύίόάήώΈΥΪΟΗΩ\d\s\?:=\.\\"\/@]*>/g;
    }
    print $mfh $_;
}
close(LCNFH) or
syserr("Unable to close licence file handle",1);

$msg=($language eq "gr"
?"Φιλικά, \nΜετεωρολογικός Σταθμός ΕΜΠ"
:"Friendly, \nWeather Station of NTUA");
print $mfh $msg;
$mfh->close or
syserr("Unable to close mail file handle",1);

```

Αυτό είναι το γράμμα για τον χρήστη. Σε αυτό του εκτυπώνουμε πάλι τις επιλογές που έκανε, για να μην ξεχάσει την μορφή του αρχείου και στο τέλος του παραθέτουμε τους όρους χορήγησης των δεδομένων.

```

$mail=new Mail::Send or
syserr("Could not create new mail object",1);
$mail->set("From",$FROM);
$msg=($language eq "gr"?"Αίτηση δεδομένων από $name $surname"

```

```

:"Data request from $name $surname");
$subject=$SUBJECT.$msg;
$mail->subject($subject);
$mail->to($TO);
$mail->set("X-Mailer", $XMAILER);
$mail->set("Content-Type", $CONTENTTYPE);
$mail->set("Content-Transfer-Encoding", $TRANSFERENCODING);
$mfh=$mail->open('sendmail') or
syserr("Can't open mail file handle",1);

```

Ξαναδημιουργούμε το στιγματότυπο \$mail και του ρυθμίζουμε τις επικεφαλίδες, για την ειδοποίηση της ομάδας του Μετεωρολογικού Σταθμού με τα στοιχεία του χρήστη που πήρε τα μετεωρολογικά δεδομένα, για στατιστικούς λόγους.

```

$msg=($language eq "gr"
?"Αγαπητέ κύριε Μετεό,\n\n".
"\tΣας στέλνω αυτό το γράμμα, για να σας ενημερώσω για την αίτηση για\n".
"χορήγηση ιστορικών δεδομένων. Ο/η χρήστης συμπλήρωσε τη φόρμα που\n".
"βρίσκεται στη σελίδα http://meteo.ntua.gr/histdata/index-\$language.html.\n".
"Τα δεδομένα της φόρμας καθώς και άλλες στατιστικές πληροφορίες είναι:\n\n"
:"Dear mister Meteo,\n\n".
"\tI send you this letter to inform you about a historical data\n".
"retrieval application. The user has filled the form located in\n".
"http://meteo.ntua.gr/histdata/index-\$language.html. The form data\n".
"and other statistical information as well are:\n\n");
print $mfh $msg;
$msg=($language eq "gr"? "Χρονικό διάστημα: $fromdate - $todate\n"
:"Time duration: $fromdate - $todate\n");
print $mfh $msg;
$msg=($language eq "gr"? "Χρονικό βήμα: $timestep\n": "Time step: $timestep\n");
print $mfh $msg;
$msg=($language eq "gr"? "Μορφή ημερομηνίας και ώρας: $dateformat\n":
"Date and time format: $dateformat\n");
print $mfh $msg;
$msg=($language eq "gr"? "Διαχωριστής στηλών: $separator\n"
:"Column separator: $separator\n");
print $mfh $msg;
$msg=($language eq "gr"? "Στήλη 1: ":"Column 1: ");
print $mfh $msg;
$msg=($language eq "gr"? "Ημερομηνία και ώρα\n": "Date and time\n");
print $mfh $msg;
for $i (0..$#series_array)
{
$item=$i+2;
$msg=($language eq "gr"? "Στήλη $item: ":"Column $item: ");
print $mfh $msg;

$sql="SELECT v.name,v.gr_name,s.firm,s.model FROM sensor s, ".
"variable v, time_series t WHERE t.series_id=$series_array[$i] and ".
"t.sensor_id=s.sensor_id and t.variable_id=v.variable_id";
$sth=$dbh->prepare($sql)
or syserr("Unable to prepare time series query:<BR>Database error: ".
"$DBI::errstr", 1);
$sth->execute
or syserr("Could not execute time series query:<BR>Database error: ".
"$DBI::errstr:<BR>\nQuery: $sql", 1);

while (@row=$sth->fetchrow_array)
{
$series_hash{$series_array[$i]}{"en"}="$row[0] $row[2] ($row[3])";
$series_hash{$series_array[$i]}{"gr"}="$row[1] $row[2] ($row[3])";
}
}
syserr("Time series fetching terminated early by error:<BR>Database ").

```

```

"error: $DBI::errstr",0) if ($DBI::err);

print $mfh $series_hash{$series_array[$i]}{$language};
print $mfh "\n";
}
$msg=($language eq "gr"? "Συμπίεση: $compress_array[$compress]\n\n"
:"Compression: $compress_array[$compress]\n\n");
print $mfh $msg;
$licencefile=$WEB_DIR."/histdata/licence-$language.html";
$msg=($language eq "gr"? "Όνομα: $name\n":"Name: $name\n");
print $mfh $msg;
$msg=($language eq "gr"? "Επώνυμο: $surname\n":"Surname: $surname\n");
print $mfh $msg;
$msg=($language eq "gr"? "Ιδιότητα: $occupation\n"
:"Occupation: $occupation\n");
print $mfh $msg;
$msg=($language eq "gr"? "Εταιρία: $company\n":"Company: $company\n");
print $mfh $msg;
$msg=($language eq "gr"? "Διεύθυνση e-mail: $email\n"
:"E-mail address: $email\n");
print $mfh $msg;
$msg=($language eq "gr"? "Σκοπός: $purpose\n\n":"Purpose: $purpose\n\n");
print $mfh $msg;
$msg=($language eq "gr"? "Μέσο χρήστη: ":"User agent: ");
print $mfh $msg;
print $mfh $query->user_agent;
print $mfh "\n";
$msg=($language eq "gr"? "Απομακρυσμένος φιλοξενητής: ":"Remote host: ");
print $mfh $msg;
$remotehost=$query->remote_host;
$remotehost=`$HOST $remotehost`;
@result_array=split /\s+/, $remotehost;
$remotehost=$result_array[$#result_array];
print $mfh "$remotehost\n";
$msg=($language eq "gr"? "Απομακρυσμένη διεύθυνση: ":"Remote address: ");
print $mfh $msg;
print $mfh $query->remote_host;
print $mfh "\n\n";
$msg=($language eq "gr"? "Φιλικά, \nΜετεωρολογικός Σταθμός ΕΜΠ":
"Friendly, \nMeteorological Station of NTUA");
print $mfh $msg;

$mfh->close or
syserr("Unable to close mail file handle",1);

```

Σε αυτό παραθέτουμε την μορφή που έχει το αρχείο που επέλεξε, τα προσωπικά του στοιχεία και διάφορες άλλες στατιστικές πληροφορίες.

```

$sth->finish
or syserr("Unable to finish cursor:<BR>Database error: $DBI::errstr", 0);

$dbh->disconnect
or syserr("Disconnect error:<BR>Database error: $DBI::errstr()", 1);
}

```

Κλείνουμε τον χειριστή εντολών και την σύνδεση στη βάση. Σε αυτό το σημείο η διεργασία παιδί για την ανάκτηση δεδομένων, έχει ολοκληρωθεί.

```

else
{
print $query->header;

```

```

$SIG{"CHLD"}="IGNORE";
$menuitem=($language eq "gr"? "istorika_dedomena": "historical_data");
open(HTMLFH,$htmlfile)
or syserr("Unable to open html file $htmlfile:<BR>System error: $!",1);

while (<HTMLFH>)
{

```

Η διεργασία πατέρας συνεχίζει και εκτυπώνει τις στάνταρ επικεφαλίδες, ρυθμίζει το βήμα που βρισκόμαστε και ανοίγει το αρχείο HTML. Όσο υπάρχουν γραμμές τις επεξεργάζεται και προσέχει τα SSIs.

```

if ($_ =~ /$INCLUDE/i)
{
    $_ =~ s/\s*<!--\s*#include\s+virtual=//i;
    $_ =~ s/\s*-->\s*//i;
    $_ =~ tr/' ' //d;
    if ($_ =~ /cgi-bin/)
    {
        open(MENUFH, "$WEB_DIR"."/menu-$language.html")
        or syserr("Unable to open menu file $_:<BR> System error: $!",2);

        while (<MENUFH>)
        {
            if ($_ =~ /$menuitem/i)
            {
                $_ =~ s#src="/buttons/menu-
$language/{menuitem}1.jpg"#src="/buttons/menu-$language/{menuitem}2.jpg"#i;
            }
            print $_;
        }
        close(MENUFH)
        or syserr("Unable to close steps file $_: $!\n",1);
    }
    else
    {
        open(INCFH, "$WEB_DIR"."/$_")
        or syserr("Unable to open include file $WEB_DIR/$_<BR>System error:
$!",2);

        while (<INCFH>)
        {
            print $_;
        }
        close(INCFH)
        or syserr("Unable to close file $WEB_DIR/$_:<BR>System error: $!",2);
    }
    next;
}

```

Σε include αν είναι CGI, ρυθμίζει το μενού με την δεύτερη εικόνα της τρέχουσας κατηγορίας, ενώ αν είναι απλό αρχείο το ενσωματώνει όπως είναι.

```

elsif ($_ =~ /$FORMDATA/i)
{
    $msg=($language eq "gr"? "Η αίτηση σας έγινε δεκτή"
: "Your application has been accepted");
    $msg=$query->b($msg);
    print $query->h2($msg);
    print $query->br;

    $msg=($language eq "gr"? "Θα σας αποσταλεί ένα e-mail με τις πληροφορίες ".
"του αρχείου, την άδεια χορήγησης και το URL που περιέχει τα δεδομένα.".

```

```

"Αυτό μπορεί να πάρει μερικά λεπτά, για αυτό θα σας παρακαλούσαμε να ".
"να μην επαναλάβετε την διαδικασία, αν δεν έχει περάσει το χρονικό ".
"διάστημα των 10 λεπτών το μέγιστο. Η διαδικασία απόκτησης ιστορικών ".
"δεδομένων έχει ολοκληρωθεί. Πατήστε στο μενού αριστερά την κατηγορία ".
"που θέλετε να πάτε."
:"You will received an e-mail with data file information, user licence ".
"and URL location. This might take several minutes, so you are kindly ".
"requested to wait and do not repeat the same procedure, ".
"unless a period of 10 minutes maximum has expired. The data retrieval ".
"procedure has been completed. You may click in one of the categories ".
"on the left menu you want to visit.");
print "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print $msg;

next;
}

```

Στην περίπτωση που είναι να εισαχθούν δεδομένα φόρμας, τότε εκτυπώνει ένα απλό μήνυμα που ενημερώνει τον χρήστη ότι η αίτησή του έγινε δεκτή.

```

print $_;
}

close(HTMLFH)
or syserr("Unable to close html file $htmlfile:<BR>System error: $!",1);
}

exit 0;

```

Όλες οι υπόλοιπες γραμμές εκτυπώνονται έτσι όπως είναι. Η διεργασία κλείνει το αρχείο HTML και τερματίζει.

Με το πρόγραμμα `sqlplus` κάναμε μία επιτυχή σύνδεση στη βάση και με την αναζήτηση όλων των εγγραφών στο view `tab`, μας φέρνει όλα τα ονόματα των πινάκων που έχουμε δημιουργήσει. Μία πιο αναλυτική περιγραφή ενός πίνακα, για παράδειγμα του πίνακα των αισθητήρων, δίνεται παρακάτω:

```
SQL> describe sensor;
```

Name	Null?	Type
SENSOR_ID	NOT NULL	NUMBER(38)
NAME		VARCHAR2(50)
STATION_ID		NUMBER(38)
GR_NAME		VARCHAR2(50)
FIRM		VARCHAR2(50)
MODEL		VARCHAR2(50)
STATUS		CHAR(1)
START_DATE		DATE
COMMENTS		VARCHAR2(256)
END_DATE		DATE
ACCURACY		VARCHAR2(10)
GR_COMMENTS		VARCHAR2(256)

Με αυτή την εντολή βλέπουμε τα πεδία που έχει ο πίνακας `sensor` και τον τύπο που είναι.

5.2.2 Πρόγραμμα `aeolus`

Το πρόγραμμα διαχείρισης `aeolus`, έχει πολλές λειτουργίες όπως είδαμε στο προηγούμενο κεφάλαιο και η σύνταξη του, καθώς και οι επιλογές του, μπορούν να φανούν καλύτερα με την παράθεση της σύντομης βοήθειας στις επιλογές και στη σύνταξη που έχει:

```
panos@acheloos:[10] ~ > aeolus help
```

```
Starting aeolus version 1.5.8 on acheloos host...
```

```
Usage: aeolus <command> [table|file|timeseries] [options]
```

Command:

```
show           for showing table record
showall        for showing all table names
add            for adding a table record
delete         for deleting a table record
update        for updating a table record
load           for loading a data file into database
make           for <normalised|hourly|daily|monthly|yearly> <n|all> series
execute        for executing an SQL statement directly to database
generate       for generating <ten-minute|hourly...> <n|all> graphs
help           for displaying a brief help
```

Table name, file or timeseries:

The table name to operate in. You can issue "showall" first to get all the available table names in the database. You can also use a data file with the load command to insert data. Time series identifier is needed on commands performing action upon them.

Options:

```
-v <yes|no>    toggling verbose mode
-s <service>   for giving the DB service to connect to
-u <user>[/<pass>] for supplying database user name and password
-p <pass>      for supplying database user password
-l <action>    for giving load action to perform (add,delete,update)
-f             force inserting values which violate unique constraints
-y            committing database changes without prompting for yes or no
```



```

-ny          aborting database changes without prompting for yes or no
-ac          enable auto commit option
-nac        disable auto commit option
-C <config> specify an alternative location for config file
-lC <loadconfig> specify an alternative location for load config
-d <no>     setting debug level (valid <no> are from 0 to 5)

```

Ένα τυπικό σενάριο λειτουργίας του, είναι η εισαγωγή, η ενημέρωση και η διαγραφή μίας εγγραφής στη βάση. Παρακάτω θα δώσουμε ένα παράδειγμα μίας τέτοιας χρήσης.

```

panos@acheloos:[12] ~ > aeolus showall
Starting aeolus version 1.5.8 on acheloos host...
Reading /etc/aeolus/aeolus.conf configuration file...
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing showall query...
Executing SQL statement...
Query statistics: 1 lines, 1 queries, 0:00:00 total time
Rate (1000 queries): 0.0000 secs, 0.0000 mins, 0.0000 hours
Total time: 0 secs, 0.0000 mins, 0.0000 hours
Printing SQL results...
Row: 'DATA_PROC'
Row: 'DATA_RAW'
Row: 'DATA_SYNTN'
Row: 'PHOTO'
Row: 'REASON'
Row: 'SENSOR'
Row: 'SENSOR_NOTAVAIL'
Row: 'SERIES_TYPE'
Row: 'STATION'
Row: 'STATION_TYPE'
Row: 'TIME_SERIES'
Row: 'TIME_STEP'
Row: 'VARIABLE'
Row: 'VIDEO'
Print statistics: 1 lines, 1 queries, 0:00:00 total time
Rate (1000 queries): 0.0000 secs, 0.0000 mins, 0.0000 hours
Total time: 0 secs, 0.0000 mins, 0.0000 hours
Summary statistics: 1 lines, 1 queries, 0:00:00 total time
Rate (1000 queries): 0.0000 secs, 0.0000 mins, 0.0000 hours
Total time: 0 secs, 0.0000 mins, 0.0000 hours
Closing database cursor...
Closing database connection...
Program aeolus finished. Exiting.

```

Στην συνέχεια θα δώσουμε ένα μικρό παράδειγμα προσθήκης, ενημέρωσης και διαγραφής ενός αισθητήρα.

```

panos@acheloos:[15] ~ > aeolus add sensor
Starting aeolus version 1.5.8 on acheloos host...
Reading /etc/aeolus/aeolus.conf configuration file...
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing add query...
Enter SENSOR_ID: 100
Enter NAME: Temp Sensor
Enter STATION_ID: 1
Enter GR_NAME: Προσωρινός Αισθητήρας
Enter FIRM: Kavalagios LTD
Enter MODEL: 200006A
Enter STATUS: W
Enter START_DATE: 01-Oct-00
Enter COMMENTS: This is a test case
Enter END_DATE:

```

```

Enter ACCURACY: 1
Enter GR_COMMENTS: Αυτή είναι μία περίπτωση δοκιμής
Executing SQL statement...
Query statistics: 1 lines, 2 queries, 0:02:11 total time
Rate (1000 queries): 65500.0000 secs, 1091.6667 mins, 18.1944 hours
Total time: 131 secs, 2.1833 mins, 0.0364 hours
Closing database cursor...
Save database changes (YES/no)?
Committing database changes...
Closing database connection...
Program aeolus finished. Exiting.

```

Το πρόγραμμα μας ζητάει είσοδο για καθένα από τα πεδία που έχει ο πίνακας στον οποίο ζητήσαμε να γίνει η εισαγωγή. Αν για κάποιο πεδίο δεν θέλουμε να βάλουμε τιμή, πατάμε enter και συνεχίζουμε. Στο τέλος μας ζητάει αν θέλουμε να σώσουμε τις αλλαγές που έχουμε κάνει στη βάση. Παρατηρούμε ότι το YES είναι με κεφαλαία, που σημαίνει ότι αν δεν δώσουμε τίποτα, εξορισμού θα θεωρήσει ότι πληκτρολογήσαμε yes. Πράγματι, έτσι είναι, αφού με το πάτημα του enter το πρόγραμμα σώζει την προσθήκη που κάναμε στη βάση.

```

panos@acheloos:[34] ~ > aeolus update sensor
Starting aeolus version 1.5.8 on acheloos host...
Reading /etc/aeolus/aeolus.conf configuration file...
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing update query...
Enter SENSOR_ID to update: 100
Update NAME ['Temp Sensor']: Stopped Sensor
Update STATION_ID ['1']:
Update GR_NAME ['Προσωρινός Αισθητήρας']: Αισθητήρας εκτός λειτουργίας
Update FIRM ['Kavalagios LTD']:
Update MODEL ['200006A']:
Update STATUS ['W']: S
Update START_DATE ['01-OCT-00']:
Update COMMENTS ['This is a test case']:
Update END_DATE ['']: 17-Oct-00
Update ACCURACY ['1']:
Update GR_COMMENTS ['Αυτή είναι μία περίπτωση δοκιμής']:
Executing SQL statement...
Query statistics: 1 lines, 3 queries, 0:01:09 total time
Rate (1000 queries): 23000.0000 secs, 383.3333 mins, 6.3889 hours
Total time: 69 secs, 1.1500 mins, 0.0192 hours
Closing database cursor...
Save database changes (YES/no)?
Committing database changes...
Closing database connection...
Program aeolus finished. Exiting.

```

Στο παραπάνω παράδειγμα, βλέπουμε ότι εκτελέσαμε το πρόγραμμα για να ενημερώσουμε την εγγραφή που μόλις βάλαμε. Του δίνουμε το αναγνωριστικό της εγγραφής και αυτό μας εμφανίζει πάλι ένα ένα τα πεδία που έχει ο πίνακας, αλλά τώρα μας δείχνει μέσα σε αγκύλες την τιμή που ήδη έχει το αντίστοιχο πεδίο. Για να αλλάξουμε μία τιμή, θα πρέπει να προσθέσουμε είσοδο στο συγκεκριμένο πεδίο.

```

panos@acheloos:[35] ~ > aeolus delete sensor
Starting aeolus version 1.5.8 on acheloos host...
Reading /etc/aeolus/aeolus.conf configuration file...
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing delete query...
Do you want to delete all tuples (yes/NO)?
Enter SENSOR_ID to delete: 100
Executing SQL statement...
Query statistics: 1 lines, 2 queries, 0:00:05 total time

```

```
Rate (1000 queries): 2500.0000 secs, 41.6667 mins, 0.6944 hours
Total time: 5 secs, 0.0833 mins, 0.0014 hours
Closing database cursor...
Save database changes (YES/no)?
Committing database changes...
Closing database connection...
Program aeolus finished. Exiting.
```

Τέλος δίνουμε και ένα παράδειγμα διαγραφής της εγγραφής που καταχωρήσαμε. Το πρόγραμμα ζητάει αν θέλει να διαγράψουμε όλες τις εγγραφές. Επειδή συνήθως αυτή δεν θα είναι η περίπτωση, η αρνητική απάντηση είναι προεπιλεγμένη. Μας ζητάει να δώσουμε το αναγνωριστικό αριθμό του αισθητήρα που θα διαγράψει. Στο τέλος μας ζητάει ξανά επιβεβαίωση, για να αποτρέψει διαγραφές που μπορεί να έγιναν από λάθος.

Το αρχείο παραμετροποίησης του προγράμματος, βρίσκεται στο `/etc/aeolus/aeolus.conf` και έχει την ακόλουθη μορφή:

```
#
# Aeolus configuration
#

# Database name
DBService="meteo.hydro.civil.ntua.gr"

# Login name
DBLogin=aeolus

# Password
DBPassword=itia

# Options
Verbose=yes
Debug=1
ForceInsert=no
SaveChanges=yes
AutoCommit=no
```

Αυτό το αρχείο περιγράφει το γενικό όνομα της βάσης δεδομένων, τον κωδικό και το σύνθημα του χρήστη της βάσης με τον οποίο θα γίνει η σύνδεση, καθώς και οι επιλογές που δέχεται το πρόγραμμα.

5.2.3 Πρόγραμμα dbonline

Το πρόγραμμα αυτόματης ενημέρωσης dbonline, έχει παρόμοιες λειτουργίες με το aeolus που περιγράφηκε προηγουμένως. Χρησιμοποιεί λίγο διαφορετική σύνταξη, μιας και η κύρια απασχόλησή του είναι η ενημέρωση της βάση δεδομένων με τις τελευταίες μετεωρολογικές μετρήσεις, τον έλεγχο τους και την μετατροπή τους σε χρονοσειρές μεγαλύτερου χρονικού βήματος.

Η σύντομη βοήθεια χρήσης του προγράμματος είναι:

```
panos@acheloos:[36] ~ > dbonline -h
Starting dbonline version 1.1.8 on acheloos host...
```

```
Usage: dbonline -f <file> [options]
```

```
-f <file>:
The file name to operate in order to insert data.
```

Options:

```
-v <yes>|<no> toggling verbose mode
-s <service> for giving the DB service to connect to
-u <user>[/<pass>] for supplying database user name and password
-p <pass> for supplying database user password
-d <no> setting debug level (valid <no> are from 0 to 5)
-y committing database changes without prompting for yes or no
-ny aborting database changes without prompting for yes or no
-C <config> specify an alternative location for config file
-lc <loadconfig> specify an alternative location for load config
-h prints this short help message
```

και μία τυπική λειτουργία του είναι η κλήση με τις ακόλουθες παραμέτρους κάθε δέκα λεπτά από το κύριο πρόγραμμα που ενημερώνει τα μετεωρολογικά δεδομένα:

```
dbonline -f /home/meteo/data/delta-t.data -lc /home/meteo/etc/delta-t.data.load
-y > /home/meteo/log/dbonline.log &
```

Το αποτέλεσμα της εκτέλεσής του δίνεται παρακάτω:

```
panos@acheloos:[6] ~ > cat /home/meteo/log/dbonline.log
Starting dbonline version 1.1.8 on acheloos host...
Reading /etc/aeolus/dbonline.conf configuration file...
Configuration file was read as following:
DBService=meteo.hydro.civil.ntua.gr (meteo.hydro.civil.ntua.gr)
DBLogin=aeolus (aeolus)
DBPassword=itia (itia)
Verbose=no (no)
Debug=3 (3)
SaveChanges=yes (yes)
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing data line...
Select query to execute in SelectData:
SELECT tname FROM tab WHERE tname='DATA_RAW'
Columns got of DATA_RAW entity: DATA_RAW
Select query to execute in SelectData:
SELECT column_name FROM all_tab_columns WHERE table_name='DATA_RAW'
Columns got of DATA_RAW entity: SERIES_ID,DATA_DATE,STATUS,VALUE
Load configuration file was read as following:
TargetTable=DATA_RAW
Action=add (add)
Delimiter=\s+
FieldNumber=14
function{Data_Date}=to_date
Got last DB record "20001017101000"
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',0.000000,to_date('20001017102000','YYYYMMDDHH24MISS'),12)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',2.215686,to_date('20001017102000','YYYYMMDDHH24MISS'),18)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',4.000000,to_date('20001017102000','YYYYMMDDHH24MISS'),19)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',0.000000,to_date('20001017102000','YYYYMMDDHH24MISS'),8)
Executing SQL statement...
```

```

Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',25.470000,to_date('20001017102000','YYYYMMDDHH24MISS'),20)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',0.621806,to_date('20001017102000','YYYYMMDDHH24MISS'),9)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',57.139200,to_date('20001017102000','YYYYMMDDHH24MISS'),11)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',19.880000,to_date('20001017102000','YYYYMMDDHH24MISS'),10)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',1285.120000,to_date('20001017102000','YYYYMMDDHH24MISS'),15)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',378.240000,to_date('20001017102000','YYYYMMDDHH24MISS'),16)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',10.000000,to_date('20001017102000','YYYYMMDDHH24MISS'),14)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',997.440000,to_date('20001017102000','YYYYMMDDHH24MISS'),13)
Executing SQL statement...
Query formed for online data insertion: INSERT INTO
DATA_RAW(Status,Value,Data_Date,Series_ID)
VALUES('U',36.174936,to_date('20001017102000','YYYYMMDDHH24MISS'),17)
Executing SQL statement...
Updating normalised, hourly, daily, monthly and yearly time series...
Making normalised ten-minute data...
Producing 101 Sunshine Duration time series: limits 0..10 and difference 10
Finished 101 Normalised time series from 1...
Producing 102 Rainfall time series: limits 0..50 and difference 50
Finished 102 Normalised time series from 2...
Producing 103 Wind Direction time series: limits 0..360 and difference 360
Finished 103 Normalised time series from 3...
Producing 104 Mean Wind Speed time series: limits 0..60 and difference 30
Finished 104 Normalised time series from 4...
Producing 105 Barometric Pressure time series: limits 890..1060 and difference
25
Finished 105 Normalised time series from 5...
Producing 106 Relative Humidity time series: limits 0..100 and difference 30
Finished 106 Normalised time series from 6...
Producing 107 Temperature time series: limits -30..60 and difference 20
Finished 107 Normalised time series from 7...
Producing 108 Rainfall time series: limits 0..50 and difference 50
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(108,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',0)
Finished 108 Normalised time series from 8...
Producing 109 Solar Radiation time series: limits -1..1500 and difference 1501
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(109,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',.621806)
Finished 109 Normalised time series from 9...
Producing 110 Temperature time series: limits -30..60 and difference 20
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(110,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',19.88)

```

```

Finished 110 Normalised time series from 10...
Producing 111 Relative Humidity time series: limits 0..100 and difference 30
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(111,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',57.1392)
Finished 111 Normalised time series from 11...
Producing 112 Rainfall time series: limits 0..50 and difference 50
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(112,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',0)
Finished 112 Normalised time series from 12...
Producing 113 Barometric Pressure time series: limits 890..1060 and difference
25
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(113,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',997.44)
Finished 113 Normalised time series from 13...
Producing 114 Sunshine Duration time series: limits 0..10 and difference 10
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(114,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',10)
Finished 114 Normalised time series from 14...
Producing 115 Total Solar Radiation time series: limits 0..3000 and difference
3000
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(115,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',1285.12)
Finished 115 Normalised time series from 15...
Producing 116 Diffuse Solar Radiation time series: limits -1..1500 and
difference 1501
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(116,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',378.24)
Finished 116 Normalised time series from 16...
Producing 117 Wind Direction time series: limits 0..360 and difference 360
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(117,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',36.174936)
Finished 117 Normalised time series from 17...
Producing 118 Mean Wind Speed time series: limits 0..60 and difference 30
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(118,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',2.215686)
Finished 118 Normalised time series from 18...
Producing 119 Wind Gust time series: limits 0..60 and difference 30
Value query formed: INSERT INTO data_proc(series_id,data_date,status,value)
VALUES(119,to_date(20001017102000,'YYYYMMDDHH24MISS'),'U',4)
Finished 119 Normalised time series from 19...
Producing 121 Wind Gust time series: limits 0..60 and difference 30
Finished 121 Normalised time series from 21...
Making normalised Hourly data...
Converting 201 Sunshine Duration time series (19991208100000-19991208104000)
Converting 202 Rainfall time series (19991208100000-19991208104000)
Converting 203 Wind Direction time series (19991208100000-19991208104000)
Converting 204 Mean Wind Speed time series (19991208100000-19991208104000)
Converting 205 Barometric Pressure time series (19991208100000-19991208104000)
Converting 206 Relative Humidity time series (19991208100000-19991208104000)
Converting 207 Temperature time series (19991208100000-19991208104000)
Converting 208 Rainfall time series (20001017100000-20001017102000)
Converting 209 Solar Radiation time series (20001017100000-20001017102000)
Converting 210 Temperature time series (20001017100000-20001017102000)
Converting 211 Relative Humidity time series (20001017100000-20001017102000)
Converting 212 Rainfall time series (20001017100000-20001017102000)
Converting 213 Barometric Pressure time series (20001017100000-20001017102000)
Converting 214 Sunshine Duration time series (20001017100000-20001017102000)
Converting 215 Total Solar Radiation time series (20001017100000-20001017102000)
Converting 216 Diffuse Solar Radiation time series (20001017100000-
20001017102000)
Converting 217 Wind Direction time series (20001017100000-20001017102000)
Converting 218 Mean Wind Speed time series (20001017100000-20001017102000)
Converting 219 Wind Gust time series (20001017100000-20001017102000)
Converting 221 Wind Gust time series (19991208100000-19991208104000)

```

```

Query statistics: 12 lines, 174 queries, 0:01:14 total time
Rate (1000 queries): 425.2874 secs, 7.0881 mins, 0.1181 hours
Total time: 74 secs, 1.2333 mins, 0.0206 hours
Closing database cursor...
Committing database changes...
Closing database connection...
Starting aeolus version 1.5.8 on acheloos host...
Connecting to meteo.hydro.civil.ntua.gr database...
Preparing generate command...
Drawing graph for 108 Ten-minute series...
Drawing graph for 109 Ten-minute series...
Drawing graph for 110 Ten-minute series...
Drawing graph for 111 Ten-minute series...
Drawing graph for 113 Ten-minute series...
Drawing graph for 114 Ten-minute series...
Drawing graph for 115 Ten-minute series...
Drawing graph for 116 Ten-minute series...
Drawing graph for 117 Ten-minute series...
Drawing graph for 118 Ten-minute series...
Drawing graph for 119 Ten-minute series...
Query statistics: 1584 lines, 68 queries, 0:00:52 total time
Rate (1000 queries): 764.7059 secs, 12.7451 mins, 0.2124 hours
Total time: 52 secs, 0.8667 mins, 0.0144 hours
Closing database connection...
Program aeolus finished. Exiting.
Aeolus statistics: 12 lines, 174 queries, 0:01:00 total time
Rate (1000 queries): 344.8276 secs, 5.7471 mins, 0.0958 hours
Total time: 60 secs, 1.0000 mins, 0.0167 hours
Summary statistics: 12 lines, 174 queries, 0:02:14 total time
Rate (1000 queries): 770.1149 secs, 12.8352 mins, 0.2139 hours
Total time: 134 secs, 2.2333 mins, 0.0372 hours
Program dbonline finished. Exiting.

```

Τα παραπάνω δεδομένα, είναι η εκτέλεση του προγράμματος για την ενημέρωση της βάσης δεδομένων με την τελευταία δεκάλεπτη μέτρηση. Το αρχείο παραμετροποίησης των δεδομένων όπως έρχονται από τον σταθμό είναι το παρακάτω:

```

#
# This is a configuration file for loading delta-t.data
#

# Target table name
TargetTable=data_raw

# What action to perform with the data (add, delete and update)
Action=add

# Value Delimiter
Delimiter=\s+

# How many values per line
FieldNumber=14

# Functions that will be used to insert custom data
Data_Date=Function[to_date]
Data_Date=Format[YYYYMMDDHH24MISS]

# Description of data fields for the add command. The '->' operator assigns
# the right value for the left field. The '#' operator declares that the number
# which follows it, is not a value, but a field position.

Data_Date=Field[0,1..13->#0]

```

```
Series_ID=Field[1->12,2->18,3->19,4->8,5->20,6->9,7->11,8->10,9->15,10->16,11->14,12->13,13->17]
Status=Field[1..13->U]
#Value=Field[1,2,3,4,5,6,7,8,9,10,11,12,13]
Value=Field[1..13]
```

Στο αρχείο παραμετροποίησης των δεδομένων, βλέπουμε ότι δηλώνεται ο πίνακας στον οποίο θα καταλήξουν. Η πράξη που θα γίνει είναι προσθήκη, ο διαχωριστής στηλών και ο αριθμός των πεδίων, ακολουθούν την δήλωση του πίνακα. Η ημερομηνία δέχεται μορφοποίηση με την `to_date` συνάρτηση και αυτό ακριβώς δηλώνεται με τις δύο γραμμές για το πεδίο `Data_Date`. Τα τελευταία πεδία δηλώνουν την θέση και τις τιμές που παίρνουν από το αρχείο δεδομένων.

5.2.4 Ιστοσελίδες

Οι σελίδες που κατασκευάστηκαν για να παρουσιάσουν τα μετεωρολογικά δεδομένα και άλλες παρόμοιες πληροφορίες, είναι στατικές αλλά και δυναμικές. Οι δυναμικές σελίδες είναι αυτές που φτιάχνουν τα CGI Perl scripts. Σε αυτή τη παράγραφο θα γίνει λεπτομερής έλεγχος στις δυναμικές σελίδες του μετεωρολογικού σταθμού, που είναι αυτές που αφορούν τα διαγράμματα και τα ιστορικά δεδομένα.

Η κεντρική σελίδα είναι αυτή που δείχνει τις τρέχουσες τιμές των μετεωρολογικών μεταβλητών.

The screenshot shows a Netscape browser window displaying the website 'Μετεωρολογικός Σταθμός ΕΜΠ - Netscape'. The address bar shows 'http://meteo.ntua.gr/index-gr.shtml'. The main content area features a banner 'Ο καιρός στο ΕΜΠ' and a navigation menu on the left with items like 'Ο καιρός τώρα', 'Στατιστικά', 'Διαγράμματα', etc. On the right, a table displays current weather data:

Τελευταία ενημέρωση	17 Oct 2000 11:20 EEST (UTC+03:00)
Θερμοκρασία	19.8°C
Υγρασία	57.1%
Μέση ταχύτητα ανέμου	2.2 m/s (2 μποφόρ, 4.3 κόμβοι)
Ριπή ανέμου	4.0 m/s
Διεύθυνση ανέμου	NE
Ατμοσφ. πίεση	
(στάθμη σταθμού)	997.4 hPa
(στάθμη θάλασσας)	1023.1 hPa
Βροχόπτωση	0.0 mm σε 10 λεπτά
Ηλιακή ακτινοβολία	621 W/m ²
Διάρκεια ηλιοφάνειας	10/10 λεπτά

At the bottom left, there is a link for 'Τομέας Υδατικών Πόρων'.

Όλες οι σελίδες έχουν ένα κεντρικό μενού στα αριστερά τους και στα δεξιά οι πληροφορίες για την κατηγορία που επιλέγεται. Όλες οι σελίδες είναι στα ελληνικά και στα αγγλικά γραμμένες. Η αγγλική κεντρική σελίδα είναι η παρακάτω:

NTUA Online Weather Data

Current conditions

Statistics

Graphs

Historical Data

General Information

Links

Photos - Video

Freq asked questions

Contact - Project team

What's new

Ελληνική έκδοση

Last update	17 Oct 2000 11:20 EEST (UTC+03:00)
Temperature	19.8°C
Humidity	57.1%
Mean wind speed	2.2 m/s (2 beaufort, 4.3 knots)
Wind gust	4.0 m/s
Wind direction	NE
Barom. pressure	(station elevation) 997.4 hPa (mean sea level) 1023.1 hPa
Rainfall	0.0 mm in 10 minutes
Solar radiation	621 W/m ²
Sunshine duration	10/10 minutes

[Department of Water Resources](#)

Στα επόμενα θα εξαιρεθούν οι αγγλικές σελίδες και θα δοθούν παραδείγματα μόνο με τις ελληνικές.

5.2.4.1 Στατιστικά

Στην κατηγορία αυτή δίνονται τα στατιστικά του τελευταίου εικοσιτετραώρου για όλες τις μετεωρολογικές μεταβλητές.

Στατιστικά - Netscape


File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/stats/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version 

Στοιχεία για το 24ωρο που τελειώνει στις 17 Oct 2000 11:20 EEST (UTC+03:00)
 Υπάρχουν δεδομένα για το 98% του 24ώρου

Γενικά

	Μέση	Ελάχιστη	Μέγιστη	Τελευταία
Θερμ (°C)	17.7	13.9 (07:20)	22.9 (15:30)	19.8
Υγρ (%)	70.9	46.0 (15:00)	90.6 (04:40)	57.1
Πίεση (hPa)	996.9	996.0 (16:20)	997.6 (10:30)	997.4
στη ΜΕΘ (hPa)	1022.8	1021.5 (16:30)	1023.5 (01:50)	1023.1

Άνεμος

	Μέση	Ελάχιστη	Μέγιστη	Τελευταία
Μέση ταχ (m/s)	1.1	0.4 (18:20)	4.0 (15:50)	2.2
Μέση τ (beaufort)	1	0	3	2
Μέση ταχ (κόμβοι)	2.2	0.8	7.8	4.3
Ριπή (m/s)	2.0	0.4 (19:00)	6.2 (15:40)	4.0

	N	NE	E	SE	S	SW	W	NW
Διεύθυνση (%)	2.1	9.2	5.6	44.4	17.6	9.2	10.6	1.4

Τρέχουσα διεύθυνση: NE

Document: Done

5.2.4.2 Διαγράμματα

Τα διαγράμματα επιλέγονται από την ομώνυμη επιλογή στο μενού που βρίσκεται αριστερά στην κεντρική σελίδα. Δίνονται όλα τα διαγράμματα που υπάρχουν με μειωμένες τις διαστάσεις τους κατά 50%, έτσι ώστε να μην επιβαρύνουν πολύ την σελίδα.

Διαγράμματα - Netscape


File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/graphs/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

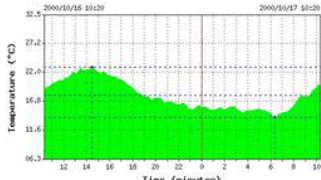
Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version 

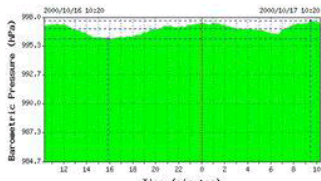
Διαγράμματα

Τα παρακάτω διαγράμματα δείχνουν τη διακύμανση των μετεωρολογικών μεταβλητών στη διάρκεια των τελευταίων 24 ωρών. Πατήστε πάνω σε ένα διάγραμμα για να το μεγεθύνετε.


[Θερμοκρασία](#)



[Βαρομετρική πίεση](#)

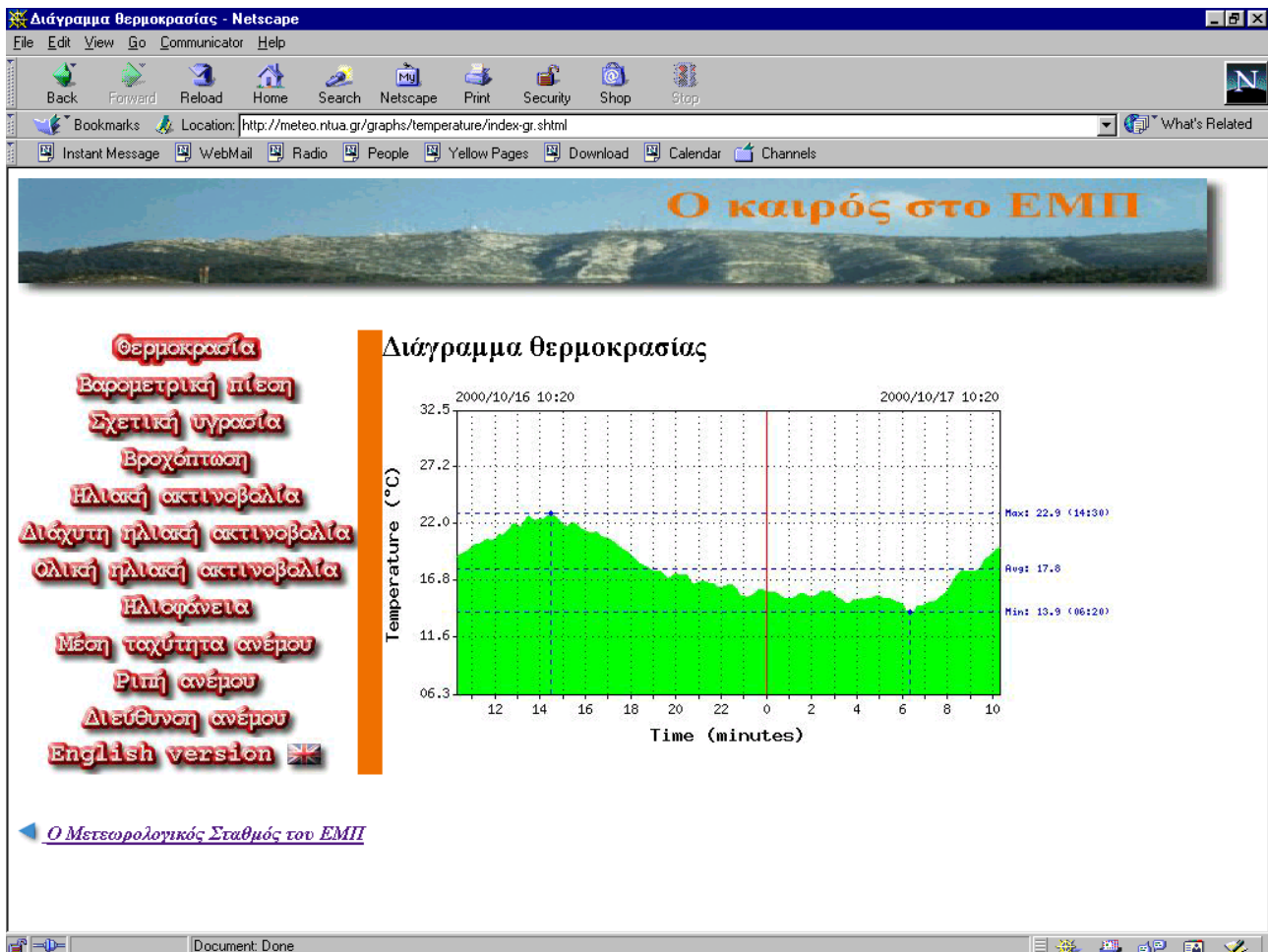


[Σχετική υγρασία](#)



Document: Done

Επιλέγοντας κάποιο από τα διαγράμματα πηγαίνουμε σε άλλο μενού και το αντίστοιχο διάγραμμα μεγεθύνεται στις κανονικές του διαστάσεις.



Στο διάγραμμα της θερμοκρασίας έχουμε στον κατακόρυφο άξονα την θερμοκρασία σε βαθμούς Κελσίου και στον οριζόντιο τον χρόνο σε λεπτά. Με οριζόντιες μπλε διακεκομμένες γραμμές, δίνονται η μέγιστη, η μέση και η ελάχιστη τιμή που είχε κατά τη διάρκεια των τελευταίων εικοσιτεσσέρων ωρών.

5.2.4.3 Ιστορικά δεδομένα

Για την απόκτηση των δεδομένων που υπάρχουν στη βάση, ο χρήστης θα πρέπει να συμπληρώσει μία αίτηση στην οποία θα προσδιορίζει ακριβώς τα δεδομένα που χρειάζεται. Αυτή ολοκληρώνεται σε 5 βήματα.

Με την επιλογή της κατηγορίας, ο χρήστης ενημερώνεται σχετικά με την κατηγορία.

Ιστορικά Δεδομένα - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/histdata/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version

Ιστορικά Δεδομένα

Αν χρειάζεστε τα μετεωρολογικά δεδομένα του σταθμού μας για κάποια μελέτη ή άσκηση, μπορείτε να τα λάβετε αντόματα συμπληρώνοντας ηλεκτρονική αίτηση. Η αίτηση είναι πολύπλοκη στη συμπλήρωσή της και απευθύνεται κυρίως σε ερευνητές ή φοιτητές. Αν θέλετε να δείτε πώς είναι ο καιρός συνήθως στην Αθήνα, δείτε τις [απαντήσεις ερωτημάτων](#) ή τα [διαγράμματα](#).

Αίτηση χορήγησης Ιστορικών Δεδομένων

[Ο Μετεωρολογικός Σταθμός του ΕΜΠ](#)

Document: Done

Με το πάτημα του κουμπιού, βρίσκεται στο πρώτο βήμα που είναι η επιλογή των μεταβλητών που τον ενδιαφέρουν.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata1.cgi> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

1. Μεταβλητές
2. Αισθητήρες
3. Μορφή αρχείου
4. Άδεια χορήγησης

1. Μεταβλητές

Προσδιορίστε τις μεταβλητές που σας ενδιαφέρουν:

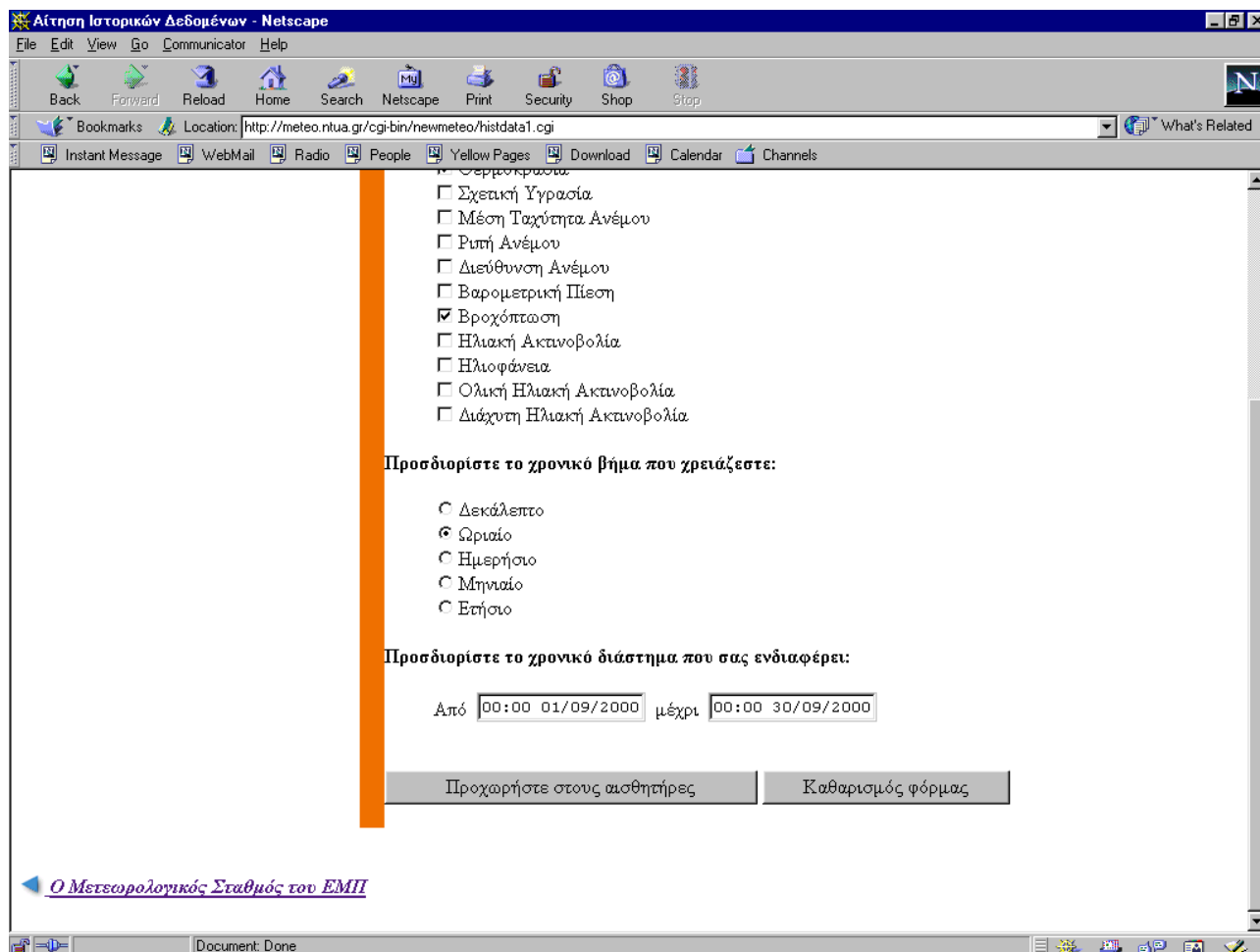
- Θερμοκρασία
- Σχετική Υγρασία
- Μέση Ταχύτητα Ανέμου
- Ριπή Ανέμου
- Διεύθυνση Ανέμου
- Βαρομετρική Πίεση
- Βροχόπτωση
- Ηλιακή Ακτινοβολία
- Ηλιοφάνεια
- Ολική Ηλιακή Ακτινοβολία
- Διάχυτη Ηλιακή Ακτινοβολία

Προσδιορίστε το χρονικό βήμα που χρειάζεστε:

- Δεκάλεπτο
- Ωριαίο
- Ημερήσιο
- Μηνιαίο
- Ετήσιο

Document: Done

Σε αυτό το βήμα έχει να επιλέξει τις μεταβλητές που θέλει να πάρει τα ιστορικά τους δεδομένα. Στο συγκεκριμένο παράδειγμα, επιλέγεται η θερμοκρασία και η βροχόπτωση.



Το χρονικό βήμα αφήνεται στο ωριαίο και το χρονικό διάστημα ρυθμίζεται για ολόκληρο το μήνα Δεκέμβριο. Όταν ολοκληρώσουμε την συμπλήρωση αυτού του βήματος, προχωράμε στους αισθητήρες.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata2.cgi> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

1. Μεταβλητές
2. Αισθητήρες
3. Μορφή αρχείου
4. Άδεια χορήγησης

2. Αισθητήρες

Προσδιορίστε τους αισθητήρες που θέλετε να πάρετε δεδομένα:

Θερμοκρασία

Αισθητήρας Θερμοκρασίας Aanderaa (3145)
 Περίοδος λειτουργίας: 13:20 30/09/1993 - 10:40 08/12/1999
 Περίοδοι ελλείψεων:
 Ακρίβεια: 1
 Παρατηρήσεις:

Αισθητήρας Θερμοκρασίας Skye Instruments (SKH 2011)
 Περίοδος λειτουργίας: 16:30 10/12/1998 -
 Περίοδοι ελλείψεων:
 Ακρίβεια: 3
 Παρατηρήσεις:

Βροχόπτωση

Αισθητήρας Βροχόπτωσης Aanderaa (3064)
 Περίοδος λειτουργίας: 13:20 30/09/1993 - 10:40 08/12/1999
 Περίοδοι ελλείψεων:
 Ακρίβεια: 02

Document: Done

Σε αυτό το βήμα εμφανίζονται όλοι οι διαθέσιμοι αισθητήρες που υπάρχουν για τις συγκεκριμένες μεταβλητές που επέλεξε ο χρήστης στο προηγούμενο βήμα. Για την θερμοκρασία επιλέγεται ο δεύτερος αισθητήρας.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata2.cgi>

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Περίοδος λειτουργίας: 10:30 10/12/1998 -

Περίοδοι ελλείψεων:

Ακρίβεια: 3

Παρατηρήσεις:

Βροχόπτωση

Αισθητήρας Βροχόπτωσης Aanderaa (3064)

Περίοδος λειτουργίας: 13:20 30/09/1993 - 10:40 08/12/1999

Περίοδοι ελλείψεων:

Ακρίβεια: 02

Παρατηρήσεις:

Αισθητήρας Βροχόπτωσης Environmental Instruments (ARG100)

Περίοδος λειτουργίας: 16:30 10/12/1998 -

Περίοδοι ελλείψεων:

Ακρίβεια: 02

Παρατηρήσεις:

Αισθητήρας Βροχόπτωσης Pronamic (Rain-O-Matic)

Περίοδος λειτουργίας: 12:10 16/11/1998 -

Περίοδοι ελλείψεων:

Ακρίβεια: 02

Παρατηρήσεις:

Προχωρήστε στη μορφή αρχείου Καθαρισμός φόρμας

[Ο Μετεωρολογικός Σταθμός του ΕΜΠ](#)

Document: Done

Για την βροχόπτωση επιλέγεται ο δεύτερος και ο τρίτος αισθητήρας. Το επόμενο βήμα είναι η μορφή του αρχείου.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata3.cgi> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

1. Μεταβλητές
2. Αισθητήρες
3. Μορφή αρχείου
4. Άδεια χορήγησης

3. Μορφή αρχείου

Το αρχείο που θα λάβετε θα είναι σε μορφή ASCII, και θα περιέχει τα δεδομένα που ζητήσατε σε στήλες. Μπορείτε παρακάτω να προσδιορίσετε τις λεπτομέρειες της μορφής του αρχείου. Η προκαθορισμένη μορφή είναι αρκετά καλή, και αν δεν θέλετε να ασχοληθείτε αφήστε την ως έχει.

Αρχείο δεδομένων

Μορφή ημερομηνίας και ώρας:

Διαχωριστής στηλών:

Στήλη 1: Ημερομηνία και ώρα

Στήλη 2:

Στήλη 3:

Στήλη 4:

Συμπίεση

Χωρίς συμπίεση

Με συμπίεση ZIP (κατάληξη .zip)

Document: Done

Σε αυτό το βήμα φαίνεται η μορφή του αρχείου. Δίνονται οι εξορισμού τιμές και η δυνατότητα του χρήστη να τις αλλάξει.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata3.cgi>

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

2. Αισθητήρες
3. Μορφή αρχείου
4. Άδεια χορήγησης

Το αρχείο που θα λάβετε θα είναι σε μορφή ASCII, και θα περιέχει τα δεδομένα που ζητήσατε σε στήλες. Μπορείτε παρακάτω να προσδιορίσετε τις λεπτομέρειες της μορφής του αρχείου. Η προκαθορισμένη μορφή είναι αρκετά καλή, και αν δεν θέλετε να ασχοληθείτε αφήστε την ως έχει.

Αρχείο δεδομένων

Μορφή ημερομηνίας και ώρα:

Διαχωριστής στηλών:

Στήλη 1: Ημερομηνία και ώρα
 Στήλη 2:
 Στήλη 3:
 Στήλη 4:

Συμπίεση

Χωρίς συμπίεση
 Με συμπίεση ZIP (κατάληξη .zip)
 Με συμπίεση GZIP (κατάληξη .gz)

[Ο Μετεωρολογικός Σταθμός του ΕΜΠ](#)

Document: Done

Στο συγκεκριμένο παράδειγμα, αλλάζουμε την μορφή της ημερομηνίας και η σειρά των στηλών άλλαξε, με την θερμοκρασία να ακολουθεί την βροχόπτωση. Επιλέγεται και η συμπίεση GZIP, αντί της ZIP.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata4.cgi> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

1. Μεταβλητές
2. Αισθητήρες
3. Μορφή αρχείου
4. Άδεια χορήγησης

4. Άδεια χορήγησης

Ζητήσατε να παραλάβετε το αρχείο με τις εξής πληροφορίες:

Χρονικό διάστημα: 00:00 01/09/2000 - 00:00 30/09/2000
Χρονικό βήμα: Ωριαίο
Μορφή ημερομηνίας και ώρας: YYYYMMDDHH24MI
Διαχωριστής στηλών: ;
Στήλη 1: Ημερομηνία και ώρα
Στήλη 2: Βροχόπτωση Environmental Instruments (ARG100)
Στήλη 3: Θερμοκρασία Skye Instruments (SKH 2011)
Στήλη 4: Βροχόπτωση Pronamic (Rain-O-Matic)
Συμπίεση: Με συμπίεση GZIP

Προσωπικές πληροφορίες:

Είναι υποχρεωτική η συμπλήρωση όλων των πεδίων. Η ηλεκτρονική σας διεύθυνση είναι αναγκαία για να σας αποσταλεί το URL με τα δεδομένα. Δεν πρόκειται στο μέλλον να σας στείλουμε τίποτε άλλο σ' αυτή τη διεύθυνση, εκτός αν προκύψει κάτι (π.χ. αν ανακαλυφθεί λάθος) σχετικά με τα δεδομένα που θα σας έχουν αποσταλεί.

Όνομα:

Επώνυμο:

Document: Done

Στο τέταρτο βήμα παρουσιάζονται όλες οι επιλογές του χρήστη που έχει κάνει μέχρι τώρα. Αν κάποια δεν του αρέσει, μπορεί να πατήσει το κουμπί back του web browser του και να επαναλάβει κάποιο από τα προηγούμενα βήματα. Στο ίδιο βήμα θα χρειαστεί ο χρήστης να βάλει μερικές προσωπικές πληροφορίες.

Αίτηση Ιστορικών Δεδομένων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/cgi-bin/newmeteo/histdata4.cgi> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Προσωπικές πληροφορίες:

Είναι υποχρεωτική η συμπλήρωση όλων των πεδίων. Η ηλεκτρονική σας διεύθυνση είναι αναγκαία για να σας αποσταλεί το URL μετά δεδομένα. Δεν πρόκειται στο μέλλον να σας στείλουμε τίποτε άλλο σ' αυτή τη διεύθυνση, εκτός αν προκύψει κάτι (π.χ. αν ανακαλυφθεί λάθος) σχετικά με τα δεδομένα που θα σας έχουν αποσταλεί.

Όνομα:

Επώνυμο:

Ιδιότητα:

Εταιρία:

Διεύθυνση e-mail:

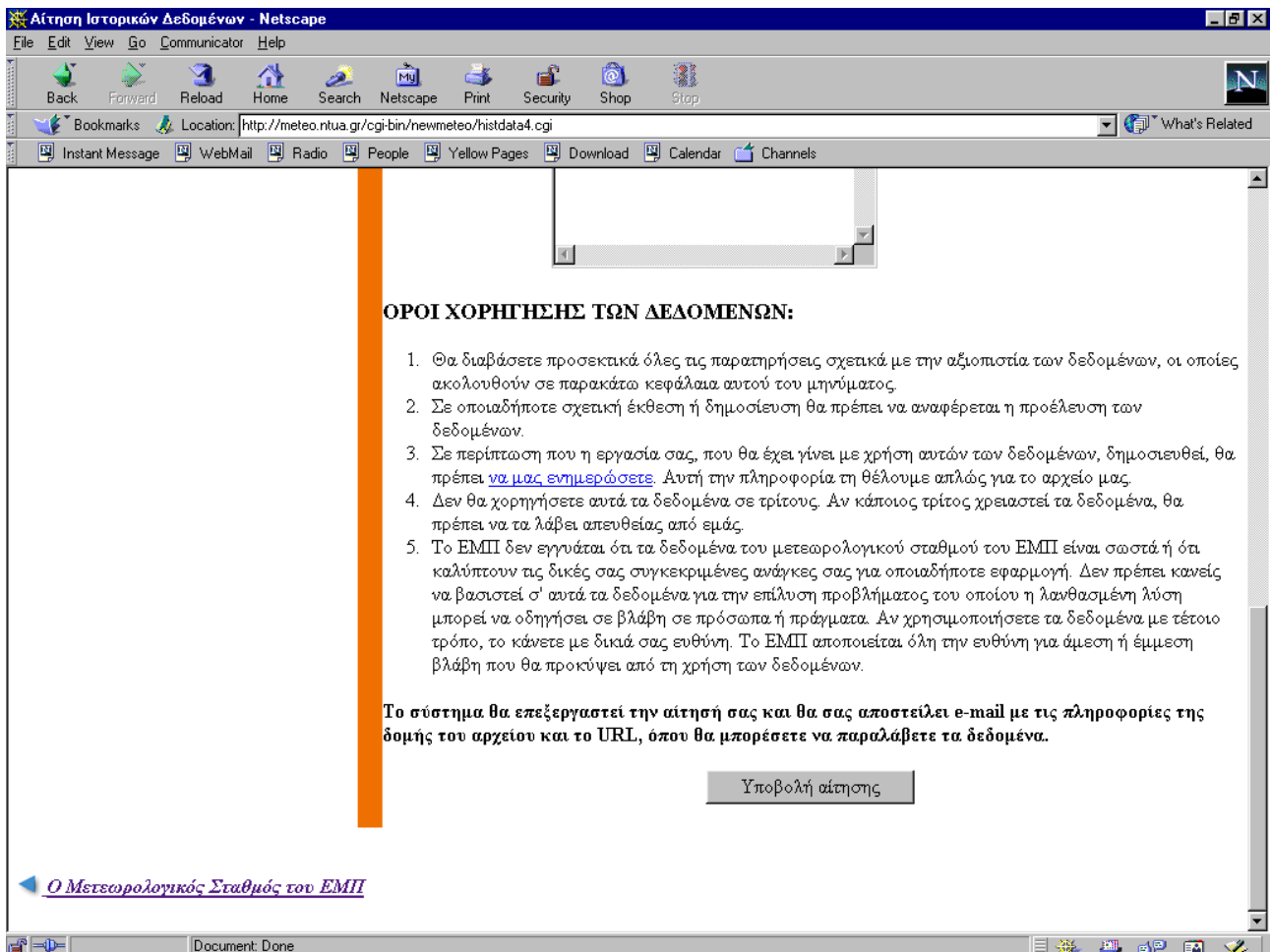
Σκοπός:

ΟΡΟΙ ΧΟΡΗΓΗΣΗΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ:

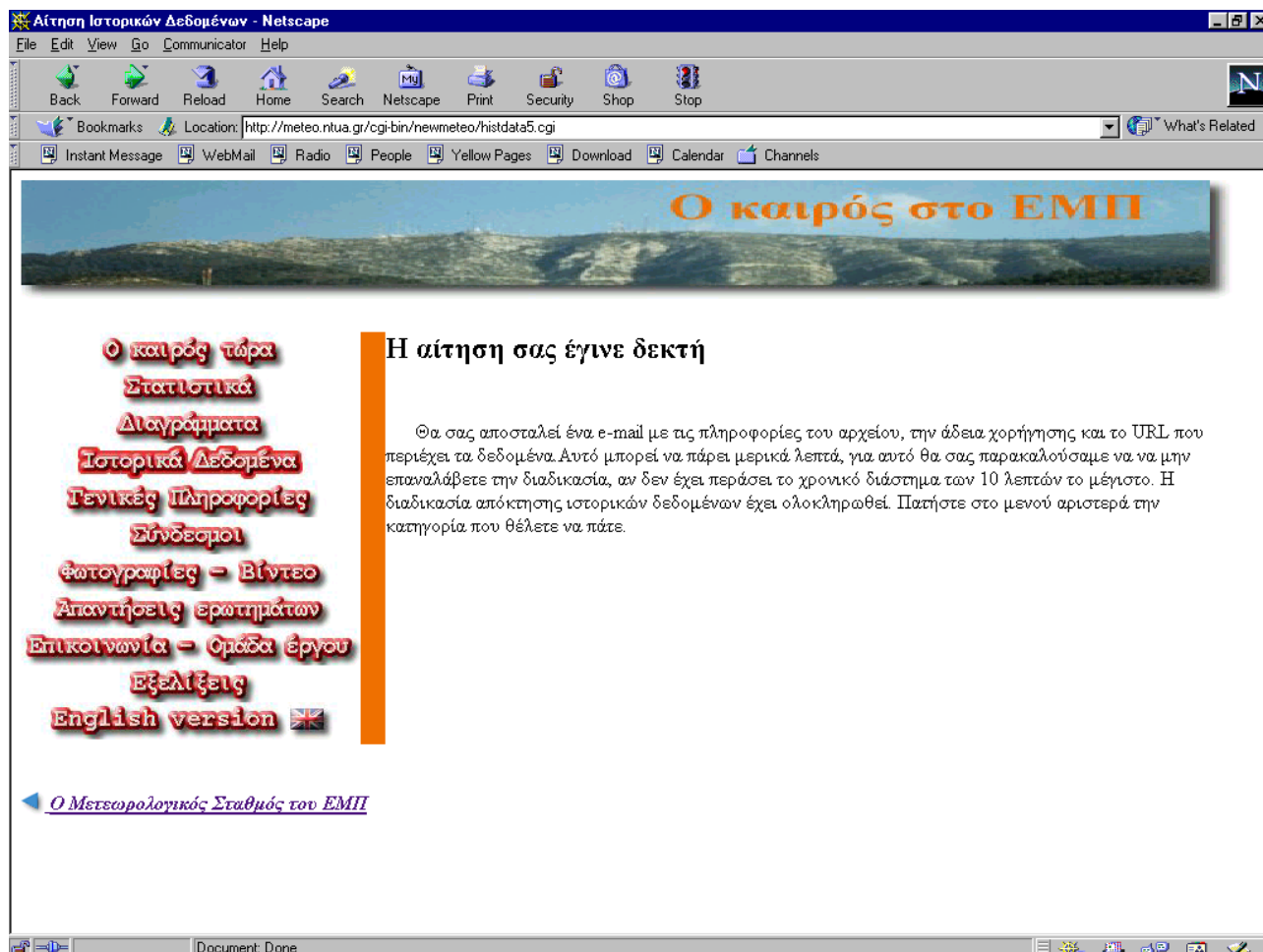
1. Θα διαβάσετε προσεκτικά όλες τις παρατηρήσεις σχετικά με την αξιοπιστία των δεδομένων, οι οποίες ακολουθούν σε παρακάτω κεφάλαια αυτού του μηνύματος.
2. Σε οποιαδήποτε σχετική έκθεση ή δημοσίευση θα πρέπει να αναφέρεται η προέλευση των

Document: Done

Οι πληροφορίες που ζητούνται είναι το ονοματεπώνυμό του, η ιδιότητα που έχει, η Εταιρία στην οποία δουλεύει, η διεύθυνση e-mail και ο σκοπός που τα χρειάζεται. Αφού τα συμπληρώσει προχωράει παρακάτω και διαβάζει τους όρους χορήγησης των δεδομένων.



Αφού ο χρήστης έχει διαβάσει τους όρους χορήγησης δεδομένων, με το πάτημα του κουμπιού προχωράει στο τελευταίο βήμα που είναι η επεξεργασία της αίτησης του χρήστη.



Στο τελευταίο βήμα, ο χρήστης ενημερώνεται ότι η αίτησή του έγινε δεκτή με επιτυχία και θα του αποσταλεί ένα ηλεκτρονικό γράμμα με πληροφορίες για το αρχείο και το από που θα πάρει τα δεδομένα του:

```
From nobody Tue Oct 17 11:44:35 2000
Return-Path: <nobody>
Received: (from nobody@localhost)
    by hydro.ntua.gr (8.9.3/8.9.3) id LAA28331;
    Tue, 17 Oct 2000 11:44:34 +0300
Date: Tue, 17 Oct 2000 11:44:34 +0300
Message-Id: <200010170844.LAA28331@hydro.ntua.gr>
Subject: METEO: Η αίτησή σας έγινε δεκτή
Content-Transfer-Encoding: 8bit
Content-Type: text/plain; charset=iso-8859-7
To: Πάνος Καβαλαγιός <P.Kavalagios@hydro.ntua.gr>
From: Online Weather Data <meteo@hydro.ntua.gr>
X-Mailer: Online Weather Mailer 1.0.7
Status: RO
```

Αγαπητέ κύριε/κυρία Πάνος Καβαλαγιός,

Η επεξεργασία των δεδομένων ολοκληρώθηκε και μπορείτε να τα παραλάβετε από την ακόλουθη διεύθυνση:

<http://meteo.ntua.gr/histdata/data/ntuameteo-200009010000-200009300000-Hourly-4.gz>

Το αρχείο που θα παραλάβετε έχει την ακόλουθη μορφή:

Χρονικό διάστημα: 00:00 01/09/2000 - 00:00 30/09/2000

Χρονικό βήμα: Ωριαίο
 Μορφή ημερομηνίας και ώρας: YYYYMMDDHH24MI
 Διαχωριστής στηλών: ;
 Στήλη 1: Ημερομηνία και ώρα
 Στήλη 2: Βροχόπτωση Environmental Instruments (ARG100)
 Στήλη 3: Θερμοκρασία Skye Instruments (SKH 2011)
 Στήλη 4: Βροχόπτωση Pronamic (Rain-O-Matic)
 Συμπύεση: Με συμπύεση GZIP

ΟΡΟΙ ΧΟΡΗΓΗΣΗΣ ΤΩΝ ΔΕΔΟΜΕΝΩΝ:

- * Θα διαβάσετε προσεκτικά όλες τις παρατηρήσεις σχετικά με την αξιοπιστία των δεδομένων, οι οποίες ακολουθούν σε παρακάτω κεφάλαια αυτού του μηνύματος.
- * Σε οποιαδήποτε σχετική έκθεση ή δημοσίευση θα πρέπει να αναφέρεται η προέλευση των δεδομένων.
- * Σε περίπτωση που η εργασία σας, που θα έχει γίνει με χρήση αυτών των δεδομένων, δημοσιευθεί, θα πρέπει να μας ενημερώσετε. Αυτή την πληροφορία τη θέλουμε απλώς για το αρχείο μας.
- * Δεν θα χορηγήσετε αυτά τα δεδομένα σε τρίτους. Αν κάποιος τρίτος χρειαστεί τα δεδομένα, θα πρέπει να τα λάβει απευθείας από εμάς.
- * Το ΕΜΠ δεν εγγυάται ότι τα δεδομένα του μετεωρολογικού σταθμού του ΕΜΠ είναι σωστά ή ότι καλύπτουν τις δικές σας συγκεκριμένες ανάγκες σας για οποιαδήποτε εφαρμογή. Δεν πρέπει κανείς να βασιστεί σ' αυτά τα δεδομένα για την επίλυση προβλήματος του οποίου η λανθασμένη λύση μπορεί να οδηγήσει σε βλάβη σε πρόσωπα ή πράγματα. Αν χρησιμοποιήσετε τα δεδομένα με τέτοιο τρόπο, το κάνετε με δικιά σας ευθύνη. Το ΕΜΠ αποποιείται όλη την ευθύνη για άμεση ή έμμεση βλάβη που θα προκύψει από τη χρήση των δεδομένων.

Φιλικά,
 Μετεωρολογικός Σταθμός ΕΜΠ

Παράλληλα με αυτό το γράμμα που έλαβε ο χρήστης, το σύστημα στέλνει και ένα γράμμα στην μετεωρολογική ομάδα meteo@meteo.ntua.gr, που ενημερώνει για την ανάκτηση δεδομένων:

```
From nobody Tue Oct 17 11:44:35 2000
Return-Path: <nobody>
Received: (from nobody@localhost)
    by hydro.ntua.gr (8.9.3/8.9.3) id LAA28334;
    Tue, 17 Oct 2000 11:44:35 +0300
Date: Tue, 17 Oct 2000 11:44:35 +0300
Message-Id: <200010170844.LAA28334@hydro.ntua.gr>
Subject: METEO: Αίτηση δεδομένων από Πάνος Καβαλαγιός
Content-Transfer-Encoding: 8bit
Content-Type: text/plain; charset=iso-8859-7
To: The Meteo Team <meteo@meteo.ntua.gr>
From: Online Weather Data <meteo@hydro.ntua.gr>
X-Mailer: Online Weather Mailer 1.0.7
Status: RO
```

Αγαπητέ κύριε Μετεό,

Σας στέλνω αυτό το γράμμα, για να σας ενημερώσω για την αίτηση για χορήγηση ιστορικών δεδομένων. Ο/η χρήστης συμπλήρωσε τη φόρμα που βρίσκεται στη σελίδα <http://meteo.ntua.gr/histdata/index-gr.html>. Τα δεδομένα της φόρμας καθώς και άλλες στατιστικές πληροφορίες είναι:

Χρονικό διάστημα: 00:00 01/09/2000 - 00:00 30/09/2000
 Χρονικό βήμα: Ωριαίο
 Μορφή ημερομηνίας και ώρας: YYYYMMDDHH24MI
 Διαχωριστής στηλών: ;
 Στήλη 1: Ημερομηνία και ώρα

Στήλη 2: Βροχόπτωση Environmental Instruments (ARG100)
 Στήλη 3: Θερμοκρασία Skye Instruments (SKH 2011)
 Στήλη 4: Βροχόπτωση Pronamic (Rain-O-Matic)
 Συμπίεση: Με συμπίεση GZIP

Όνομα: Πάνος
 Επώνυμο: Καβαλαγιός
 Ιδιότητα: Φοιτητής
 Εταιρία: ΕΜΠ
 Διεύθυνση e-mail: P.Kavalagios@hydro.ntua.gr
 Σκοπός: Ανάκτηση δεδομένων για
 θερμοκρασία και βροχόπτωση
 για τον μήνα Σεπτέμβριο.

Μέσο χρήστη: Mozilla/4.73 [en] (Win98; I)
 Απομακρυσμένος φιλοξενητής: rpp116.dialup.ntua.gr
 Απομακρυσμένη διεύθυνση: 147.102.223.116

Φιλικά,
 Μετεωρολογικός Σταθμός ΕΜΠ

Σε αυτό το γράμμα δίνονται οι επιλογές που έκανε ο χρήστης για να πάρει δεδομένα, τις πληροφορίες που έβαλε στην αίτηση και μερικές στατιστικές πληροφορίες για τον web browser που χρησιμοποίησε και την διεύθυνση IP και το host name που προέρχεται.

5.2.4.4 Γενικές πληροφορίες

Σε αυτή την κατηγορία, δίνονται διάφορες γενικές πληροφορίες για τον σταθμό.

Γενικές πληροφορίες - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/geninfo/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version

Ο αυτόματος μετεωρολογικός σταθμός του ΕΜΠ

Τοποθεσία
 Πολυτεχνειούπολη Ζαγράφου, Αθήνα, περίπου 4.5 km ανατολικά της Πλατείας Συντάγματος, στους δυτικούς πρόποδες του Υμηττού.
[Χάρτης](#)

Συντεταγμένες
 37°58'26" N, 23°47'16" E, υψόμετρο εδάφους 219 m

Μονάδες καταγραφής
 Μία μονάδα Ananderaa για τους παλιούς ψηφιακούς αισθητήρες, με τοπική μονάδα αποθήκευσης. Οι αισθητήρες διαβάζονται κάθε δέκα λεπτά, κατόπιν σήματος που δίνεται από χρονόμετρο ενσωματωμένο στη μονάδα. Οι μετρήσεις μεταβιβάζονται αφενός σε ραδιοπομπό και αφετέρου σε τοπική μονάδα αποθήκευσης. Η λήψη των δεδομένων γίνεται από το Εργαστήριο, 1 km μακριά, από PC, μέσω του ραδιοδέκτη.

Μία μονάδα Delta-T για τους νέους αναλογικούς αισθητήρες, με ενσωματωμένη μνήμη για αποθήκευση δεδομένων, και ενσωματωμένο ρολόι. Οι αισθητήρες διαβάζονται ανά 5 λεπτά (περίοδος δειγματοληψίας) και οι μετρήσεις τους αποθηκεύονται ανά 10 λεπτά. Η επικοινωνία γίνεται μέσω σειριακής θύρας, modem και καλωδίου με Linux PC στο Εργαστήριο, 1 km μακριά.

Αισθητήρες
 Αναλογικοί:

- Υγρασία και θερμοκρασία αέρα
- Βροχόπτωση

Document: Done

5.2.4.5 Σύνδεσμοι

Σε αυτή την κατηγορία δίνονται διάφοροι άλλοι σύνδεσμοι σε σελίδες με μετεωρολογικό περιεχόμενο.

The screenshot shows a Netscape browser window with the title 'Σύνδεσμοι - Netscape'. The address bar shows the URL 'http://meteo.ntua.gr/links/index-gr.shtml'. The main content area features a banner with the text 'Ο καιρός στο ΕΜΠ' over a landscape image. Below the banner is a vertical navigation menu on the left with items like 'Ο καιρός τώρα', 'Στατιστικά', 'Διαγράμματα', 'Ιστορικά Δεδομένα', 'Γενικές Πληροφορίες', 'Σύνδεσμοι', 'Φωτογραφίες - Βίντεο', 'Απαντήσεις ερωτημάτων', 'Επικοινωνία - Ομάδα έργου', 'Εξελίξεις', and 'English version'. The main content area has a section titled 'Σύνδεσμοι' with a sub-section 'Προγνώσεις' containing a paragraph and a list of links: 'Yahoo Weather', 'CNN interactive', 'Intellicast', 'USA TODAY', and 'Washington Post'. Below this is a section 'Παρόμοιες ελληνικές σελίδες' with two bullet points. At the bottom, there is a section 'Διάφορα' with one bullet point.

Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version

Σύνδεσμοι

Προγνώσεις

Στις παρακάτω θέσεις θα βρείτε μετεωρολογικές προγνώσεις για την Αθήνα. Αν νάξετε σ' αυτές τις ιστοσελίδες, θα βρείτε επίσης προγνώσεις για όλα τα μέρη του κόσμου.

- [Yahoo Weather](#)
- [CNN interactive](#)
- [Intellicast](#)
- [USA TODAY](#)
- [Washington Post](#)

Επίσης υπάρχει και το δελτίο καιρού της [Εθνικής Μετεωρολογικής Υπηρεσίας](#).

Παρόμοιες ελληνικές σελίδες

- Μια άλλη ερευνητική ομάδα στο ΕΜΠ διαθέτει παρόμοιο μετεωρολογικό σταθμό, του οποίου οι μετρήσεις επίσης [διατίθενται στο δίκτυο](#).
- Το Εθνικό Αστεροσκοπείο Αθηνών [διαθέτει στο δίκτυο δεδομένα](#) από το μετεωρολογικό σταθμό Πεντέλης (μονο αγγλικά).

Διάφορα

- Στο [WWW Virtual Library: Meteorology](#) θα βρείτε πολλούς συνδέσμους σε σελίδες σχετικές με τον

5.2.4.6 Φωτογραφίες – Βίντεο

Φωτογραφίες ή βίντεο για τον σταθμό, δίνονται σε αυτή την κατηγορία.

Φωτογραφίες - Βίντεο - Netscape


File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: <http://meteo.ntua.gr/photos/index-gr.shtml> What's Related


Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

- Ο καιρός τώρα
- Στατιστικά
- Διαγράμματα
- Ιστορικά Δεδομένα
- Τεχνικές Πληροφορίες
- Σύνδεσμοι
- Φωτογραφίες - Βίντεο
- Απαντήσεις ερωτημάτων
- Επικοινωνία - Ομάδα έργου
- Εξελίξεις
- English version 

Φωτογραφίες του σταθμού

Λήφθηκαν στις 5 Ιανουαρίου 1998.



Ανατολική άποψη του σταθμού. Δεξιά είναι ο ιστός με όλα τα στοιχεία του σταθμού. Αριστερά είναι ένας μετεωρολογικός κλωβός με συμβατικά όργανα, και πίσω του ένα συμβατικό εξαιμιόμετρο. Στη μέση ένας συμβατικός βροχογράφος. Η κύρια χρήση των συμβατικών οργάνων είναι να επιδεικνύονται στους φοιτητές, αλλά ενίοτε χρησιμοποιούνται για τον έλεγχο των μετρήσεων των αισθητήρων.

Document: Done

5.2.4.7 Απαντήσεις ερωτημάτων

Σε αυτή την κατηγορία δίνονται απαντήσεις σε διάφορα συνηθισμένα ερωτήματα που γίνονται από τους χρήστες.

Απαντήσεις ερωτημάτων - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/faq/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

Απαντήσεις σε ερωτήματα για τον καιρό στο ΕΜΠ

- [Κάνετε προγνώσεις;](#)
- [Θα ήθελα να έχω πρόγνωση του καιρού για μετά από ένα μήνα.](#)
- [Σκοπεύω να επισκεπτώ την Αθήνα και θα ήθελα να ξέρω πώς είναι συνήθως ο καιρός αυτή την εποχή.](#)
- [Θα πάω για ιστιοπλοΐα ή για wind surfing. Μπορείτε να μου δώσετε πληροφορίες για τους ανέμους;](#)
- [Έχετε δεδομένα για άλλες θέσεις στην Ελλάδα, ή γνωρίζετε πού μπορώ να βρω;](#)
- [Το σύστημα δείχνει ότι η θερμοκρασία έπεσε ξαφνικά από τους 20°C στους -13°C, ή ότι κατά το τελευταίο εικοσιτετράωρο η ελάχιστη θερμοκρασία ήταν -13°C.](#)
- [Διαβάζω πως οι αρχαίοι Αθηναίοι φορούσαν μόνο χιτόνα και μιάσι, ακόμα και το χειμώνα. Δεν φαίνεται να είναι ένδυση που να προστατεύει αρκετά από το κρύο. Μήπως έχει αλλάξει το κλίμα από τότε;](#)

Κάνετε προγνώσεις;

Ο σταθμός μας απλά καταγράφει πώς είναι ο καιρός τώρα και στο παρελθόν, και δεν μπορεί να κάνει πρόγνωση για το μέλλον. Μόνο η Εθνική Μετεωρολογική Υπηρεσία κάνει προγνώσεις. Στη [σελίδα συνδέσμων](#) θα βρείτε συνδέσμους σε θέσεις που παρουσιάζουν την πρόγνωση του καιρού της Αθήνας για τις επόμενες 4 ή 5 μέρες.

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Γενικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα Έργου
Εξελίξεις
English version

Document: Done

5.2.4.8 Επικοινωνία – Ομάδα Έργου

Αν κάποιος θέλει να επικοινωνήσει με την μετεωρολογική ομάδα του σταθμού, μπορεί να το κάνει αντλώντας τις πληροφορίες που χρειάζεται από αυτή την κατηγορία.

Επικοινωνία - Ομάδα έργου - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://meteo.ntua.gr/contact/index-gr.shtml> What's Related

Instant Message WebMail Radio People Yellow Pages Download Calendar Channels

Ο καιρός στο ΕΜΠ

Ο καιρός τώρα
Στατιστικά
Διαγράμματα
Ιστορικά Δεδομένα
Τεχνικές Πληροφορίες
Σύνδεσμοι
Φωτογραφίες - Βίντεο
Απαντήσεις ερωτημάτων
Επικοινωνία - Ομάδα έργου
Εξελίξεις
English version

Επικοινωνία

Ερωτήσεις, σχόλια και υποδείξεις είναι ευπρόσδεκτες. Πριν κάνετε ερωτήσεις βεβαιωθείτε ότι έχετε διαβάσει τις [απαντήσεις των συχνών ερωτημάτων](#). Η ηλεκτρονική μας διεύθυνση είναι

meteo@meteo.ntua.gr

Ο Καιρός ΕΜΠ είναι έργο του [επ. καθ. Δημήτρη Κουτσογιάννη](#). Ο [Νίκος Μανάσης](#) είναι υπεύθυνος για τη λειτουργία και τα δεδομένα του σταθμού. Ο [Γιάννης Χριστοφίδης](#) και ο [Παναγιώτης Κεβαλαγιός](#) είναι υπεύθυνοι για το λογισμικό και τις ιστοσελίδες.

Κατά καιρούς πολλοί άνθρωποι συνέβαλαν στο έργο. Ο Κ. Κωνσταντίνιδης εγκατέστησε το σταθμό και βοήθησε όποτε υπήρχε βλάβη. Ο Νάσος Παπακώστας δημιούργησε την πρώτη υπηρεσία finger με δεδομένα καιρού, ενώ και πρόσφατα προσέφερε πολύτιμη τεχνική βοήθεια σε θέματα τηλεπικοινωνίας. Ο Παναγιώτης Χριστιάς ανέπτυξε την πρώτη έκδοση των ιστοσελίδων καθώς και τα διαγράμματα. Η Ράνα Τσουμάνη, τότε φοιτήτρια, πραγματοποίησε μετρήσεις με συμβατικά όργανα και συγκρίσεις με τις ενδείξεις των αισθητήρων. Ο Κ. Κουριδάκης κατασκεύασε τη διαμόρφωση του χώρου και της περίφραξης. Ο Δ. Κουβιάς της SCIENTACT ΑΕ εγκατέστησε τη νέα μονάδα καταγραφής και τα νέα όργανα και βοήθησε σε διάφορα τεχνικά ζητήματα. Η Άννα Πατρικίου επέβλεψε την ανάπτυξη των νέων σελίδων σε θέματα αισθητικής. Η [Γιάννα Σταματάκη](#) ετοίμασε τους χάρτες θέσης του σταθμού. Ο Αντρέας Σακελλαρίου και ο Αλέξανδρος Μανώτας πρόσφεραν πρόσθετη υπολογιστική υποστήριξη. Ευχαριστούμε ακόμα την Κωνσταντίνα Σακκά, υπεύθυνη τηλεφωνικού δικτύου του ΕΜΠ, για την κατανόηση και τις συμβουλές της, και γενικά όλο το Κέντρο Δικτύων ΕΜΠ για την υποστήριξή του σε θέματα τηλεπικοινωνιών.

[Τομέας Υδατικών Πόρων](#)

Document: Done

5.2.4.9 Εξελίξεις

Οι αλλαγές που γίνονται στον σταθμό, καταγράφονται σε αυτή τη σελίδα, με πρώτη κάθε φορά την πιο πρόσφατη εξέλιξη.

Εξελίξεις

Τελευταία ενημέρωση: 17 Μαΐου 2000

Νέοι αισθητήρες, βελτιώσεις στο σύστημα, και βίντεο, Απρίλιος-Μάιος 2000

Προστέθηκαν νέοι αισθητήρες ταχύτητας και διεύθυνσης ανέμου, και νέος αισθητήρας ηλιακής ακτινοβολίας που μετρά συνολική, απευθείας και από διάχυση ηλιακή ακτινοβολία. Η επικοινωνία είναι πλέον αποκλειστικά ενσύρματη με μεγάλη αξιοπιστία. Οι σελίδες ανανεώνονται γρηγορότερα. Δείτε τη νέα [σελίδα στατιστικών](#), το νέο [διάγραμμα ηλιακής ακτινοβολίας](#).

Νέα μονάδα καταγραφής, αισθητήρες, και ενσύρματη επικοινωνία

Από το φθινόπωρο του 1998 έγιναν σταδιακά αλλαγές στο σταθμό. Ο σταθμός απέκτησε νέα μονάδα καταγραφής και νέους αισθητήρες. Η νέα μονάδα επικοινωνεί με το εργαστήριο με κατευθείαν καλώδιο και modem. Η επικοινωνία είναι αμφίδρομη, με αποτέλεσμα να μπορεί να ελεγχθεί η πληροφορία και να διορθώνονται όλα τα σφάλματα μετάδοσης. Ακόμα, η διαχείριση της μονάδας καταγραφής μπορεί να γίνεται από το εργαστήριο. Η τροφοδοσία του σταθμού γίνεται πλέον από το ηλεκτρικό δίκτυο, με μπαταρίες για την περίπτωση διακοπής. Οι νέοι αισθητήρες μετρούν θερμοκρασία, υγρασία, βροχόπτωση και ηλιακή ακτινοβολία. Έτσι, οι παλιοί αισθητήρες υγρασίας και βροχόπτωσης, που έχουν βλάβη, καταργούνται εντελώς, ενώ για τη θερμοκρασία υπάρχουν πλέον δύο αισθητήρες, που κάνει δυνατή την πραγματοποίηση συγκρίσεων.

Οι προτιμήσεις των επισκεπτών για τη μορφή των διαγραμμάτων

Σε αυτό το σημείο ολοκληρώθηκε και ο έλεγχος για τις ιστοσελίδες του σταθμού. Στο επόμενο κεφάλαιο θα δοθεί ο επιλογος αυτής της διπλωματικής εργασίας με συμπεράσματα και μελλοντικές επεκτάσεις που μπορεί να πάρει το σύστημα.

6 ΚΕΦΑΛΑΙΟ 6: ΕΠΙΛΟΓΟΣ

6.1 Σύνοψη και συμπεράσματα

Το σύστημα εισαγωγής, επεξεργασίας και παρουσίασης των μετεωρολογικών δεδομένων του αυτόματου μετρητικού σταθμού, οργανώθηκε σε σχεσιακή βάση δεδομένων. Η βάση από μόνη της προσφέρει μόνο άρτια οργάνωση. Για την υλοποίηση των τριών σταδίων, χρησιμοποιήθηκαν τα προγράμματα aeolus και dbonline, το πρώτο για διαχείριση και γραφήματα και το δεύτερο για αυτόματη ενημέρωση. Στο στάδιο της παρουσίασης επίσης χρησιμοποιήθηκε το World Wide Web και CGI Perl Scripts.

Τα συμπεράσματα που βγάλαμε από αυτή την υλοποίηση, καλύπτουν τους στόχους που είχε το καινούργιο σύστημα.

- Άρτια οργάνωση
- Ευέλικτη στατιστική επεξεργασία
- Εύκολη πρόσβαση σε όλα τα ιστορικά δεδομένα

6.2 Μελλοντικές επεκτάσεις

Το καινούργιο σύστημα παρέχει πραγματικά απεριόριστες δυνατότητες και χωράει αρκετές επεκτάσεις. Τα στατιστικά είναι δυνατό να βγουν για οποιοδήποτε χρονικό διάστημα, εύκολα και γρήγορα. Μπορεί να είναι δίνεται και σαν επιλογή στο χρήστη να καθορίσει αυτός το χρονικό διάστημα που θέλει να δει ένα συγκεκριμένο στατιστικό στοιχείο. Για παράδειγμα θα μπορούσε να πάρει το άθροισμα της βροχόπτωσης για κάποιον συγκεκριμένο μήνα στο παρελθόν.

Το ίδιο επίπεδο επέκτασης δέχονται και τα διαγράμματα των μετεωρολογικών μεταβλητών. Το χρονικό διάστημα μπορεί εύκολα να επεκταθεί από τις τελευταίες 24 ώρες, σε ένα οποιοδήποτε χρονικό διάστημα στο παρελθόν. Έτσι ο χρήστης θα μπορεί να έχει την δυνατότητα να πάρει την γραφική παράσταση της θερμοκρασίας για όλους του καλοκαιρινούς μήνες για να παρατηρήσει εύκολα τον θερμότερο μήνα.

Μία τελευταία επέκταση, είναι η αναζήτηση ομοίων καταστάσεων στο παρελθόν. Με αυτό το μοντέλο, θα επιλέγονται μία ή περισσότερες μετεωρολογικές μεταβλητές και δοθέντος του τρέχοντος χρονικού παραθύρου, θα γίνεται αναζήτηση στο παρελθόν της βάσης για να εκλεχθεί μία καμπύλη που να προσεγγίζει περισσότερο την σημερινή. Όταν βρεθεί, γνωρίζουμε ήδη την εξέλιξή της και μπορεί να δοθεί σαν πιθανή εξέλιξη της σημερινής.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

Για την συγγραφή της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκαν πληροφορίες από τις ακόλουθες πηγές:

- [ΑνΓι93] Αναδρανιστάκης Μ. και Χ. Γιαννακός, Ανάπτυξη μεθόδων ποιοτικού ελέγχου χωροχρονικά σύμφωνα με τα διεθνή πρότυπα, Υδροσκόπιο-Τεύχος 5/6, 1993
- [Colb98] Rafe Colburn, CGI Programming, 1998
- [Γαλλ95] Άγγελος Γαλλής, Προγραμματίστε σε HTML, 1995
- [ScCh97] Randal L. Schwartz, Tom Christiansen, Learning Perl, 1997
- [SiSP99] Nathan Patwardhan, Perl in a nutshell, 1999

Εκτός από αυτά τα βιβλία, χρησιμοποιήθηκε η τεκμηρίωση που προσφέρει η Oracle για την βάση δεδομένων της και το resource repository:

<http://metalink.oracle.com>

καθώς και η τεκμηρίωση και το FAQ της γλώσσας προγραμματισμού Perl:

<http://www.perl.com/pub/v/documentation>

<http://www.perl.com/pub/v/faqs>