

## ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία, αναπτύσσεται ένα εσωτερικό δίκτυο (intranet) μιας πραγματικής εταιρείας. Στόχος είναι η έγκυρη και γρήγορη διασπορά των πληροφοριών σε όλες τις βαθμίδες των στελεχών της εταιρείας καθώς και η θεμελίωση της υποδομής του πληροφοριακού της συστήματος για εφαρμογές ηλεκτρονικού εμπορίου και επέκτασης του δικτύου της σε extranet. Για το σκοπό αυτό, χρησιμοποιούμε την ανερχόμενη τεχνολογία του διαδικτύου, την Extended Markup Language μερικές επεκτάσεις της οποίας είναι ακόμη υπό σχεδιασμό από το W3C. Ωστόσο, τα σημαντικά πλεονεκτήματα της εν λόγω τεχνολογίας, η επιθυμία μας να χρησιμοποιήσουμε τεχνολογίες αιχμής αλλά και η πεποίθηση ότι η τεχνολογία XML θα κυριαρχήσει στο εγγύς μέλλον, μάς ώθησαν στη χρήση της. Στην εργασία που ακολουθεί, παρουσιάζονται με λεπτομέρειες, εφαρμογές της XML, XSL, XQL, XML DOM, XML Schemata σε συνδυασμό με την τεχνολογία asp και μία σχεσιακή βάση δεδομένων. Το αποτέλεσμα είναι η δυνατότητα υποβολής ερωτημάτων σε μία βάση δεδομένων, η χρήση ενεργών καναλιών, η αναζήτηση στο δικτυακό τόπο, η άμμεση στατιστική ενημέρωση και η εφαρμογή μιας περίπτωσης Extranet.

## Abstract

In this thesis, we develop an intranet of an existing company. The development of this intranet aims to the rapid dissemination of information to all management levels considering the potential expanding of the company's intranet to an extranet. We use, as main constructing tool the Internet's evolving technology, the extended markup language. Although some of it's extensions are not totally specified yet, by the W3C, the major advantages of XML and our belief that it is going to be Internet's main technology, pushed us to use it as a cutting edge technology. Below, we are going to present in detail, XML, XSL, XQL, XML DOM and XML Schemata application in combination with active server pages technology and a relational database. The final result will be the ability of requesting information from the database, usage of active channels, a search engine, statistical information in graphic format and a paradigm of an Extranet application.

## ΠΡΟΛΟΓΟΣ

Αναμφισβήτητη η επιστήμη της πληροφορικής έδωσε σχετικά πρόσφατα σε μικρότερου μεγέθους επιχειρήσεις, την ευκαιρία να καρπωθούν τις πολλές δυνατότητες που προσφέρει. Η δυνατότητα επεξεργασίας της πληροφορίας, η δημιουργία νέου τύπου πληροφοριών και η εύκολη πρόσβαση των ενδιαφερόμενων σε μεγάλους και επεξεργασμένους όγκους πληροφοριών δίνει σήμερα τη δυνατότητα για πιο αποτελεσματικό και πολυεπίπεδο management καθώς και λύσεις σε προβλήματα τόσο ρουτίνας όσο και στρατηγικού επιπέδου.

Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με μία πραγματική επιχείρηση (εισηγμένη στο Χ.Α.Α.) που υπάγεται στις μεσαίου μεγέθους βιομηχανικές και εμπορικές επιχειρήσεις. Ο εκρηκτικός ρυθμός ανάπτυξης και επέκτασης της εν λόγω επιχείρησης σε μικρό χρονικό διάστημα, είχε ως φυσικό επακόλουθο προβλήματα που αφορούν στην οργάνωση του υπάρχοντος πληροφοριακού συστήματος. Η επιχείρηση διαθέτει περί τα 35 τερματικά σε δίκτυο Ethernet των 100 Mbps ενώ ο server είναι σε Windows NT με SQL server ver.6.5, Internet Information Server 4.0, τα οποία περικλείονται στο Microsoft BackOffice Small Business Server .

Η δραστηριότητα της εταιρείας είναι η παραγωγή και εμπορία εξαρτημάτων υδραυλικών ανελκυστήρων και απασχολεί περίπου 250 άτομα μεταξύ των οποίων και 30 περίπου μηχανολόγοι μηχανικοί.

Ο στόχος της εργασίας μας θα είναι να διευκολύνουμε τη διάδοση των πληροφοριών εντός της εταιρείας με νέες τεχνολογίες που θα μπορούν να προσαρμοστούν και για χρήση στο web έτσι ώστε επιλεγμένοι συνεργάτες της να έχουν πρόσβαση σε επιλεγμένου τύπου πληροφορίες αλλά και για να είναι έτοιμη να εισέλθει στο ηλεκτρονικό εμπόριο όταν αυτό διαδοθεί στις τάξεις των πελατών αλλά και των προμηθευτών της. Έτσι, θα μπορεί να αυξηθεί η παραγωγικότητα σε εργασίες ρουτίνας, η ποιότητα και η ταχύτητα λήψης αποφάσεων με άμεση επίδραση στην αποτελεσματικότητα αλλά και στην αποδοτικότητα της εταιρείας. Επίσης παρατηρήθηκε μεγάλος όγκος πλεονάζουσων πληροφοριών και εφαρμογών, γεγονός το οποίο μπορεί να αντιμετωπιστεί με τη λειτουργία ενός αποτελεσματικού intranet.

Η έννοια Intranet ορίζεται ως εξής : [<http://www.whatis.com/intranet.htm>]  
“ Ένα Intranet είναι ένα ιδιωτικό δίκτυο μιας επιχείρησης ή οργανισμού. Μπορεί να αποτελείται από πολλά Local Area Networks και χρησιμοποιεί μισθωμένες γραμμές σε ένα Wide Area Network. Συνήθως, ένα Intranet

επικοινωνεί με το Internet μέσω ενός ή περισσότερων gateways. Ο κύριος σκοπός του είναι να διαμοιράσει τις πληροφορίες μιας εταιρείας και τούς υπολογιστικούς πόρους, στους εργαζόμενους. Μπορεί επίσης να χρησιμοποιηθεί για να διευκολύνει την ομαδική εργασία και τις τηλεδιασκέψεις.

Ένα intranet χρησιμοποιεί TCP/IP, HTTP και άλλα πρωτόκολλα του Internet. Συνήθως οι μεγάλες εταιρείες αφήνουν εξωτερικούς χρήστες να έχουν πρόσβαση στα intranets, μέσω firewalls έτσι ώστε να διασφαλίζεται η ακεραιότητα του intranet. Όταν ένα μέρος του Intranet μπορεί να προσεγγιστεί από πελάτες, προμηθευτές, κ.τ.λ., εκείνο το κομμάτι του δικτύου γίνεται Extranet. “

Αναλυτικότερα, μετά το πέρας της εργασίας, θα έχουμε παρουσιάσει εφαρμογές του Intranet που καλύπτουν τις εξής περιπτώσεις :

- ✓ Δυνατότητα εύρεσης στοιχείων εμβόλου για κάποια συγκεκριμένη παραγγελία
- ✓ Στοιχεία παραγγελίας για κάποιο συγκεκριμένο πελάτη
- ✓ Ανεύρεση εμβόλων από αποθέματα με χρήση κριτηρίων μήκους
- ✓ Λήψη παραγγελιών ανά μήνα
- ✓ Πωλήσεις σασι ανά είδος σε συγκεκριμένο χρονικό διάστημα
- ✓ Πωλήσεις δοχείων ανά μήνα
- ✓ Έλεγχος παραγγελιών κάποιου πελάτη σε συγκεκριμένο χρονικό διάστημα
- ✓ Στατιστικά πωλήσεων ανά πόλη
- ✓ Έλεγχος των οικονομικών στοιχείων των πελατών με γραφική σήμανση
- ✓ Έλεγχος αποθεμάτων με γραφική σήμανση
- ✓ Προβολή τεχνικών στοιχείων δοχείων
- ✓ Πλοήγηση στο αρχείο εργαζόμενων
- ✓ Εφαρμογή Extranet με δυνατότητα απομακρυσμένου χρήστη να παρακολουθεί τα οικονομικά του στοιχεία και χρήση κωδικού
- ✓ Τήρηση στατιστικών παραπόνων και βλαβών
- ✓ Δυνατότητα ενημέρωσης αρχείου εμβόλων αποθεμάτων μέσω HTTP
- ✓ Χρήση ενεργών καναλιών
- ✓ Εσωτερικός μηχανισμός αναζήτησης

Οι παραπάνω περιπτώσεις καλύπτουν τα περισσότερα είδη πληροφοριών που το Intranet της εν λόγω επιχείρησης θα πρέπει να

παρέχει. Επιδεικνύοντας τον κώδικα και τη μεθοδολογία, η δημιουργία νέων εφαρμογών καθίσταται πολύ εύκολη για τον προγραμματιστή, αρκεί να γνωρίζει τις βασικές αρχές της XML, Vbscript και βάσεων δεδομένων.

Τα βασικά προϊόντα της εταιρείας χωρίζονται στις παρακάτω τρεις βασικές κατηγορίες :

*Έμβολο* (πιστόνι που περιλαμβάνει έμβολο και κύλινδρο καθώς και κάποια μικροεξαρτήματα)

*Δοχείο* (ο ανυψωτικός μηχανισμός που εμπεριέχει καζάνι, κινητήρα, μπλοκ βαλβίδων και αντλία)

*Σασί* (ο μηχανισμός στον οποίο επικάθεται ο θάλαμος επιβατών)

Οποιαδήποτε άλλα προϊόντα μπορούν να συμπεριληφθούν στα παραπάνω ή μπορούν να αποτελέσουν ξεχωριστές ομάδες αλλά αυτό δεν επηρεάζει καθόλου τις μεθόδους που θα ακολουθήσουμε.

**Σημείωση : Η γλώσσα XML προϋποθέτει την ύπαρξη του Internet Explorer 5.0 και άνω.** Κανένας άλλος κατασκευαστής δεν έχει εκδώσει φυλλομετρητή συμβατό με τα standards του W3C έως σήμερα (Ιανουάριος 2000).

Επίσης χρησιμοποιήθηκαν demo εκδόσεις των παρακάτω εργαλείων ανάπτυξης λογισμικού XML :

XML spy (<http://www.XMLspy.com>)

XML Writer της Wattle software (<http://XMLwriter.net>)

## ΕΙΣΑΓΩΓΗ

Η διάρθρωση της εργασίας έχει ως εξής :

Στο κεφάλαιο 1 παρουσιάζουμε συνοπτικά τη σχεσιακή βάση δεδομένων που αναπτύχθηκε έτσι ώστε να καταχωρούνται τα στοιχεία που μας ενδιαφέρουν ή να μεταφέρονται από άλλα προγράμματα και να καταχωρούνται στη βάση.

Στο κεφάλαιο 2 παρουσιάζεται η δομή των εγγράφων μας σε XML και η παραγωγή αυτών μέσω της τεχνολογίας asp (active server pages). Η παρουσίαση αυτή εξελίσσεται από τον μικρότερο στο μεγαλύτερο βαθμό δυσκολίας και πολυπλοκότητας.

Στο κεφάλαιο 3 παρουσιάζονται xsl stylesheets τα οποία και αντιστοιχούν στα προαναφερθέντα XML έγγραφα (documents)

Στο κεφάλαιο 4 παρουσιάζονται τα XML schemata μέσω των οποίων γίνεται ο έλεγχος εγκυρότητας (validation ) των XML εγγράφων.

Στο κεφάλαιο 5, όλα τα παραπάνω συνδέονται στο τελικό αποτέλεσμα, όπου και γίνεται και η παρουσίαση του. Παρουσιάζονται ολοκληρωμένες εφαρμογές, οι οποίες μπορούν να αποτελέσουν ολοκληρωμένο οδηγό για ανάπτυξη XML εγγράφων και των επεκτάσεών της, από τον αναγνώστη.

Τέλος στο κεφάλαιο 6, αξιολογούμε το αποτέλεσμα που προκύπτει με τη χρήση της XML και καταλήγουμε σε συμπεράσματα και περαιτέρω προτάσεις.

Θα πρέπει να διευκρινίσουμε ότι στην παρούσα εργασία σκοπός μας είναι να δείξουμε τους τρόπους με τους οποίους μπορεί να χρησιμοποιηθεί η τεχνολογία XML σε συνδυασμό με άλλες τεχνολογίες και όχι να φτάσουμε σε ένα τελικό αποτέλεσμα το οποίο θα τεθεί σε εφαρμογή άμεσα, καθώς κάτι τέτοιο μπορεί να είναι εφικτό έπειτα από λεπτομερή ανάλυση των απαιτήσεων της εταιρείας. Έτσι, θα παρουσιάσουμε διάφορες εναλλακτικές λύσεις για να εξάγουμε παρόμοια αποτελέσματα. Με αυτόν τον τρόπο, όταν η εταιρεία θα αποφασίσει να θέσει σε λειτουργία το προτεινόμενο intranet, θα μπορεί να επιλέξει ανά περίπτωση κάποιες από τις μεθόδους που προτείνουμε.

## ΚΕΦΑΛΑΙΟ 1

### Σχεσιακή βάση δεδομένων

Το RDMS (Relational DataBase Management System) που προτιμήθηκε για την ανάπτυξη της βάσης δεδομένων ήταν η Microsoft Access 97. Οι αιτίες της επιλογής αυτής ήταν πρώτον το ότι η εταιρεία διαθέτει το εν λόγω λογισμικό και δεύτερον η άψογη συνεργασία της με τον υπάρχοντα SQL server. Επίσης, το παραπάνω εργαλείο είναι εύχρηστο και διαθέτει πολλά βοηθήματα ανάπτυξης.

**Σημείωση : Τα στοιχεία που ακολουθούν (πελατών, τεχνικά και της εταιρείας), δεν είναι πραγματικά.**

Αρχικά σχεδιάσαμε τους παρακάτω πίνακες :

*Customers* : Πίνακας με τους ενεργούς πελάτες της εταιρείας και στοιχεία που τους αφορούν.

*Block*: Όλοι οι τύποι των μπλοκ βαλβίδων που χρησιμοποιεί η εταιρεία.

*Diair* : Οι ιδιότητες που μπορεί να έχει ένα έμβολο (π.χ. μονοκόμματο, διαιρούμενο κ.τ.λ).

*Doxeio* : Όλοι οι τύποι καζανιών.

*Employees* : Τα τμήματα πωλήσεων της εταιρείας.

*Emvola* : Όλοι οι τύποι εμβόλων που κατασκευάζει η εταιρεία.

*Motor* : Όλοι οι τύποι κινητήρων που χρησιμοποιεί η εταιρεία.

*Oikonomika* : Οικονομικά στοιχεία των πελατών (η ενημέρωση γίνεται από το λογιστικό πακέτο που χρησιμοποιεί η εταιρεία).

*ORDERS* : Ο βασικός πίνακας παραγγελιών με το σημαντικότερο κλειδί (τον αριθμό παραγγελίας)

*Pistons* : Ομαδοποίηση των εμβόλων βάση της διατομής τους.

*Pump* : Οι αντλίες που χρησιμοποιούνται.

*Rails* : Είδος οδηγών (απαραίτητο τεχνικό χαρακτηριστικό για τη διεκπεραίωση μιας παραγγελίας).

*Rodes* : Υποσύνολο του παρακάτω πίνακα, τεχνικό χαρακτηριστικό του σασί.

*Sassi* : Όλα τα πιθανά είδη σασί.

*Stocks*: Αποθέματα εμβόλων για άμεση παράδοση (ενημερώνεται από βάση του τμήματος παραγωγής).

*Τεχνικά* : Τεχνικά στοιχεία της παραγγελίας.

*Τροχ* : Όλες οι τροχαλίες που χρησιμοποιούνται.

*Τυγος* : Όλοι οι τύποι αναρτήσεων ενός υδραυλικού ανελκυστήρα.

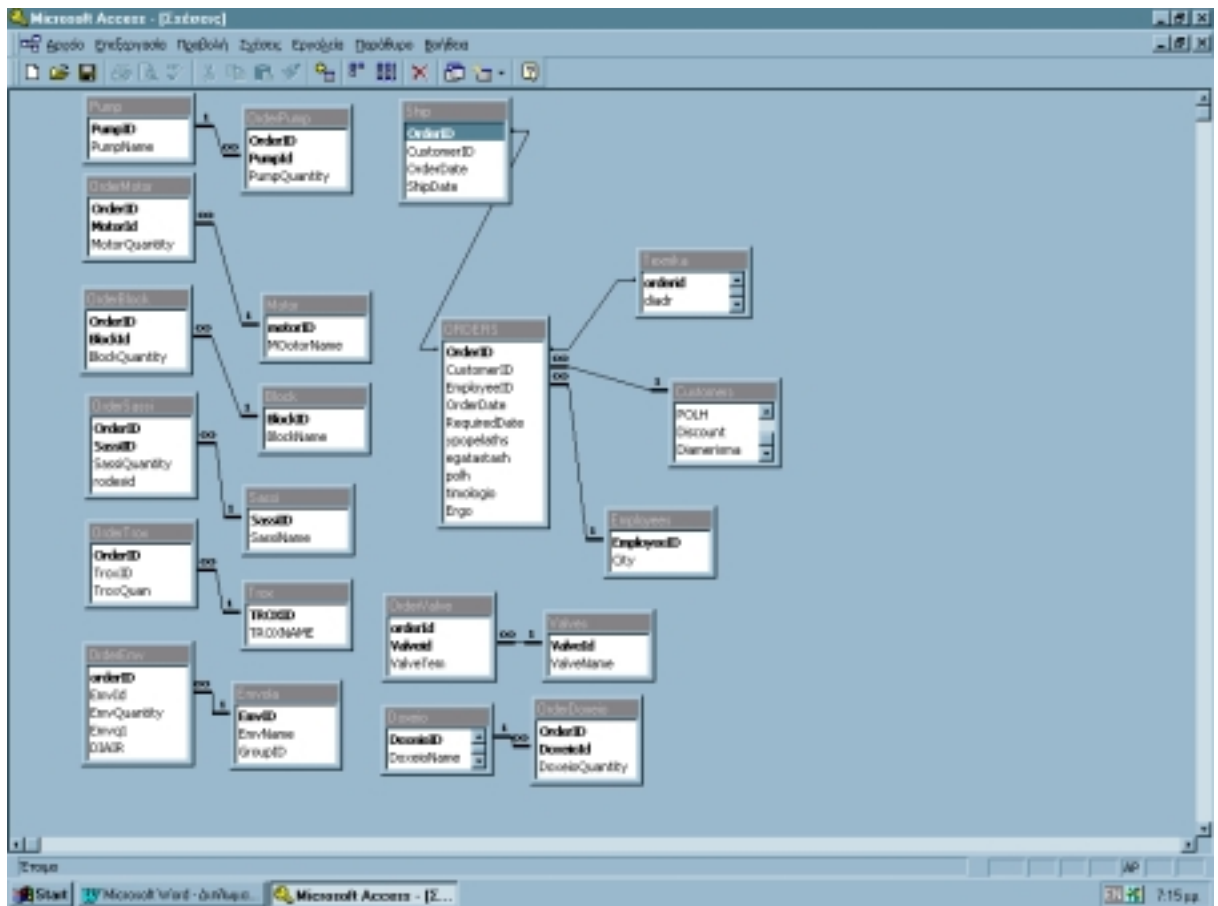
*Valves* : Βαλβίδες, υποσύνολο των εμβόλων.

*Ware* : Υλικά και αποθέματά τους

Οι παρακάτω πίνακες είναι οι πίνακες που περιλαμβάνουν τα στοιχεία κάθε εξαρτήματος για κάθε παραγγελία.

OrderBlock, OrderDoxeio , OrderEmv , OrderEx , OrderMotor, OrderPump, OrderSassi, OrderTrox..

Οι σχέσεις μεταξύ των πινάκων δημιουργήθηκαν αφού κανονικοποιήσαμε τη βάση δεδομένων (orders.mdb) και απεικονίζονται παρακάτω :



**Εικόνα 1**

Επίσης δημιουργήθηκε μια φόρμα εισαγωγής παραγγελιών με το όνομα «Παραγγελίες».

**Εικόνα 2**

Θα πρέπει να σημειώσουμε ότι για την ανάκτηση δεδομένων μέσω των μεθόδων που θα χρησιμοποιήσουμε, δεν είναι απαραίτητη η δημιουργία ερωτημάτων, αφού θα χρησιμοποιηθεί η SQL στα active server pages που θα δημιουργήσουμε.

Επίσης, για την επιτυχία τέτοιων εφαρμογών όπως αυτές που ακολουθούν, είναι πολύ σημαντικό να ακολουθηθεί η διαδικασία της κανονικοποίησης (normalization).

Ως κανονικοποίηση ορίζεται η «διαδικασία αντιστοίχισης ιδιοτήτων σε οντότητες (entities). Η κανονικοποίηση μειώνει τα περιττά δεδομένα και συνεπώς εξαλείφει τις ανωμαλίες που μπορούν να προκληθούν από τέτοιου είδους δεδομένα. Η κανονικοποίηση δεν καταργεί πλήρως τα περιττά



δεδομένα αλλά ελέγχει το βαθμό περιπλοκότητας έτσι ώστε να μπορούμε να συνδέουμε σωστά τους πίνακες των βάσεων δεδομένων.»

[Rob P., Coronel C., Database Systems, 3<sup>rd</sup> edition, ITP, 1997, σελ.282]

Δε θα αναφερθούμε περαιτέρω στη σχεσιακή βάση δεδομένων καθώς δεν αποτελεί το πρωτεύων αντικείμενο της παρούσας εργασίας και επίσης διότι τα αποτελέσματά μας θα ήταν ακριβώς τα ίδια αν είχαμε χρησιμοποιήσει οποιαδήποτε άλλο RDBMS.

## ΚΕΦΑΛΑΙΟ 2

### Μετατροπή των δεδομένων σε XML

Όπως έχουμε προαναφέρει, σκοπός της εργασίας μας είναι η διάχυση πληροφοριών μέσω του πρωτοκόλλου http σε ένα Intranet αλλά και στο web (με κάποιες προσθήκες και μετατροπές). Οι μέχρι πρόσφατα πιο ενδεδειγμένες λύσεις ήταν η χρήση της γλώσσας HTML με την προσθήκη κάποιων scripts (σε Java, Vbscript κ.τ.λ.)

Στο παρών όμως έργο, επιλέχθηκε μια νέα και πολλά υποσχόμενη τεχνολογία, η eXtended Markup Language ή XML.

#### Παρουσίαση XML

Η γλώσσα (ή σωστότερα μετα-γλώσσα –meta language-) XML είναι ένα υποσύνολο του προτύπου SGML. Η μεγαλύτερη διαφορά μεταξύ HTML και XML είναι ότι η HTML είναι μια γλώσσα που περιγράφει κυρίως την εμφάνιση μιας σελίδας. Παραδείγματος χάριν, το tag <H2> είναι ένα headline που απεικονίζεται σε ένα συγκεκριμένο μέγεθος. Η XML αντίθετα δεν περιγράφει τη μορφή μιας σελίδας αλλά τι είναι οι λέξεις ή τα δεδομένα που βρίσκονται σε ένα έγγραφο. Στην XML π.χ. μπορούμε να ορίσουμε ένα tag με το όνομα <First\_Name> και σε αυτό το tag να περικλείονται ονόματα. Ενώ δηλαδή η HTML συνδυάζει δομή και εμφάνιση, η XML τα διαχωρίζει. Αυτό σημαίνει ότι τα έγγραφα σε XML μπορούν να χρησιμοποιηθούν πιο εύκολα σε διάφορα είδη εφαρμογών.

Για να γίνουν τα παραπάνω πιο κατανοητά, παραθέτουμε μερικούς ορισμούς της XML :

« Η XML είναι μια markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Οι δομημένες πληροφορίες μπορούν να περιέχουν όλα τα είδη πληροφοριών (λέξεις, εικόνες κ.λ.π.) και κάποιες ενδείξεις για το ρόλο αυτών των πληροφοριών (π.χ., το περιεχόμενο μιας επικεφαλίδας έχει διαφορετική σημασία από το περιεχόμενο ενός υποσελίδου). Σχεδόν όλα τα έγγραφα έχουν κάποια δομή.

Μία markup γλώσσα είναι ένας μηχανισμός για να αναγνωρίζουμε τη δομή σε ένα έγγραφο (document). Το πρότυπο της XML ορίζει έναν συγκεκριμένο τρόπο να προσθέτουμε markup σε ένα έγγραφο.»

[Walsh N., What is XML?, <http://www.xml.com/xml/pub/98/10/guide1.html>]

« Η XML είναι ένας ευέλικτος τρόπος δημιουργίας κοινής μορφοποίησης (format) και διαμοιρασμό format και δεδομένων στο Word Wide Web, σε Intranets και αλλού. Παραδείγματος χάριν, οι κατασκευαστές ηλεκτρονικών υπολογιστών μπορεί να συμφωνήσουν σε μία τυποποίηση ή σε έναν κοινό τρόπο περιγραφής ενός προϊόντος και μετά να περιγράψουν τις πληροφορίες του προϊόντος με XML. Ένας τέτοιος προκαθορισμένος τρόπος περιγραφής δεδομένων, θα επέτρεπε σε κάποιον χρήστη να στείλει έναν “intelligent agent” στα Web sites κάθε κατασκευαστή υπολογιστών, να μαζέψει στοιχεία και να τα συγκρίνει για να βρεί π.χ. τη φθηνότερη μνήμη.»

[<http://whatis.com/xml.htm>]

Οι ειδικοί της Intel, αξιολόγησαν την XML ως εξής :

« Η XML αναδύεται σαν ένας ισχυρός μηχανισμός ανταλλαγής δεδομένων για επιχειρησιακές εφαρμογές και για το Web. Η XML είναι μια κυρίαρχη τεχνολογία όχι μόνο για τις χρησιμοποιούμενες πληροφορίες μέσα στην επιχείρηση αλλά και για τις συναλλακτικές δυνατότητες μεταξύ των επιχειρήσεων μέσω του Internet και επιχειρησιακών Intranets.»

[[http://realtimes.com/rtnews/rtipages/19990226\\_whatisxml.htm](http://realtimes.com/rtnews/rtipages/19990226_whatisxml.htm)]

Παρακάτω παρουσιάζουμε συνοπτικά μερικούς λόγους που ενισχύουν την επιλογή μας :

- ✓ Όλα τα HTML έγγραφα είναι ήδη και XML. Η XML μπορεί να ορίσει εκδοχές της HTML.
- ✓ Ορίζοντας τη δική μας markup γλώσσα μπορούμε να προγραμματίσουμε με μεγαλύτερη ακρίβεια και σαφήνεια.
- ✓ Δείχνει πολύ καλύτερα τη δομή ενός εγγράφου άρα διευκολύνει την πλοήγηση και την ανεύρεση.
- ✓ Επιτρέπει σε ένα έγγραφο να απεικονίζεται με πολλούς τρόπους (μέσω XSL ή CSS).
- ✓ Οι δυνατότητες διασύνδεσης υπερκειμένου (hypertext linking) είναι πολύ μεγαλύτερες από τις αντίστοιχες της HTML.
- ✓ Είναι ανεξάρτητη από πλατφόρμες
- ✓ Η XML μειώνει το φόρτο των δικτύων. Ένα ερώτημα το οποίο τίθεται σε HTML και τα αποτελέσματά του διαβιβάζονται στον πελάτη (client) δε μπορεί να επεξεργαστεί, ενώ στην XML μπορούμε π.χ. να συγκεκριμενοποιήσουμε περαιτέρω το ερώτημα. Στην HTML αντίθετα πρέπει να θέσουμε νέο ερώτημα.

✓ Είναι ευρέως διαδεδομένο ότι η XML είναι το επόμενο βήμα στην εξέλιξη του web.

Ένα έγγραφο XML πρέπει να είναι σωστά μορφοποιημένο (well formed), αλλιώς ο parser δεν το προβάλλει. Τα tags της XML είναι case sensitive και όταν ανοίγει κάποιο tag πρέπει απαραίτητα και να κλείνει. Επίσης, τα στοιχεία που έχουν σχέση γονέα – παιδιού (nested) θα πρέπει να είναι δομημένα σωστά ώστε το έγγραφο να είναι well formed.

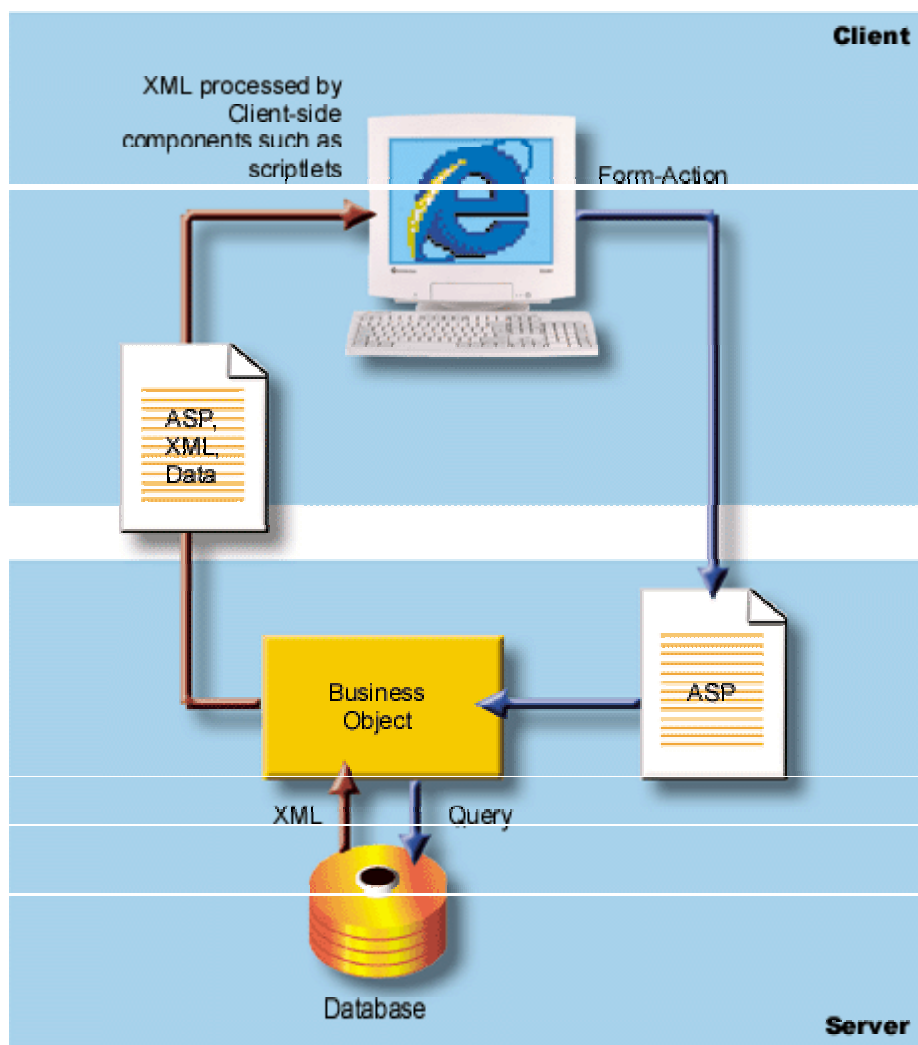
Για τις απλές σελίδες ωστόσο προτείνουμε HTML αφού έχει μια σειρά πλεονεκτημάτων όπως το ότι δεν είναι case sensitive, δεν είναι απαραίτητο να είναι well formed κ.τ.λ.

### **Μετατροπή δεδομένων**

Όπως έχουμε προαναφέρει τα δεδομένα που θέλουμε να παρουσιάσουμε στο δίκτυό μας βρίσκονται σε πίνακες ή ερωτήματα της βάσης δεδομένων. Για να τα μετατρέψουμε σε XML θα χρησιμοποιήσουμε την τεχνολογία ASP (Active Server Pages). Η επεξεργασία των δεδομένων θα γίνεται στον server έτσι ώστε αν αποφασιστεί το intranet να γίνει extranet, να μην υπάρχουν προβλήματα με τους φυλλομετρητές που ενδεχομένως να μην είναι I.E 5.0.

Το μοντέλο που θα χρησιμοποιήσουμε είναι ευρέως διαδεδομένο για την ανάπτυξη Intranets και Extranets χωρίς την προσθήκη της XML. Το καινοτόμο σημείο της εργασίας δηλαδή, είναι η προσθήκη της XML και των επεκτάσεών της, κάτι το οποίο τώρα αρχίζει να εφαρμόζεται και στο εξωτερικό, όπως τουλάχιστον έχει διαπιστωθεί από την πλοήγησή μας στον παγκόσμιο ιστό και τη συμμετοχή μας σε σχετικές ηλεκτρονικές λίστες.

Μια πρώτη γραφική αναπαράσταση της δομής του Intranet που στοχεύουμε να καταλήξουμε είναι η ακόλουθη :



**Εικόνα 3**

Η βάση δεδομένων από την οποία αντλούμε στοιχεία πρέπει να δηλωθεί στο ODBC (Open DataBase Connectivity) των Windows. Έτσι, δηλώνουμε τη βάση δεδομένων orders στο System DSN (Data Source Name) έτσι ώστε να είναι προσεγγίσιμη από όλους τους χρήστες αλλά και τις υπηρεσίες των NT. Όπως φαίνεται και από την παραπάνω εικόνα, κάποιος απομακρυσμένος υπολογιστής θέτει ένα ερώτημα στον server, μέσω μιας φόρμας από την οποία παίρνει τα στοιχεία που ο αιτών θέλει, ένα αρχείο asp. Από τη βάση δεδομένων δίνονται στοιχεία σε XML τα οποία συγκροτούν ένα έγγραφο το οποίο μπορεί να περιέχει XML στοιχεία, asp και άλλου είδους δεδομένα.

### Λίγα λόγια για τον IIS 4.0 (Internet Information Server)

Ο IIS 4.0 παρέχει κάποια χρήσιμα αντικείμενα (objects) τα οποία διευκολύνουν την πρόσβαση σε ODBC βάσεις δεδομένων. Τα βασικότερα από αυτά είναι τα ακόλουθα :

- **Request** – παίρνει πληροφορίες από τον χρήστη
- **Response** – στέλνει πληροφορίες στον χρήστη
- **Server** – ελέγχει τον Internet Information Server
- **Application** – κατανέμει πληροφορίες επιπέδου εφαρμογής και ελέγχει τις ρυθμίσεις για όσο χρονικό διάστημα διαρκεί η εφαρμογή

Μέσω των scripts, θα δημιουργούμε κάθε φορά ένα Active Data Object μέσω του οποίου θα συνδεόμαστε στη βάση.

Τα πλεονεκτήματα αυτής της μεθόδου είναι ότι :

- ✓ Δημιουργούμε XML δεδομένα χωρίς πολύ κόπο ακόμα και από μεγάλες βάσεις δεδομένων
- ✓ Η XML παράγεται δυναμικά από τα δεδομένα της βάσης κι έτσι όταν ενημερώνεται η βάση παίρνουμε XML έγγραφα, πλήρως ενημερωμένα.

### Three Tier Architecture

Για την ανάπτυξη των εφαρμογών μας, επιλέξαμε μία γνωστή αρχιτεκτονική, την Three Tier Architecture. Η βασική ιδέα αυτής της αρχιτεκτονικής είναι ο διαχωρισμός της σε τρία επίπεδα που επικοινωνούν και αλληλεπιδρούν μεταξύ τους. Τα τρία αυτά επίπεδα είναι τα εξής :

- ✓ **Data Services** (Παραγωγή του εγγράφου XML)

Τα δεδομένα μιας εφαρμογής βρίσκονται στο πρώτο επίπεδο. Η XML είναι μια πολύ καλή λύση αφού αποτελεί ένα κοινό format για διάφορους τύπους αποθήκευσης δεδομένων. Τα data services εφαρμόζουν την παραγωγή της δομημένης XML από τα υπάρχοντα δεδομένα.

[βλ. βιβλιογραφία Νο 4, σελ. 388]

- ✓ **Business services** (Data Integration)

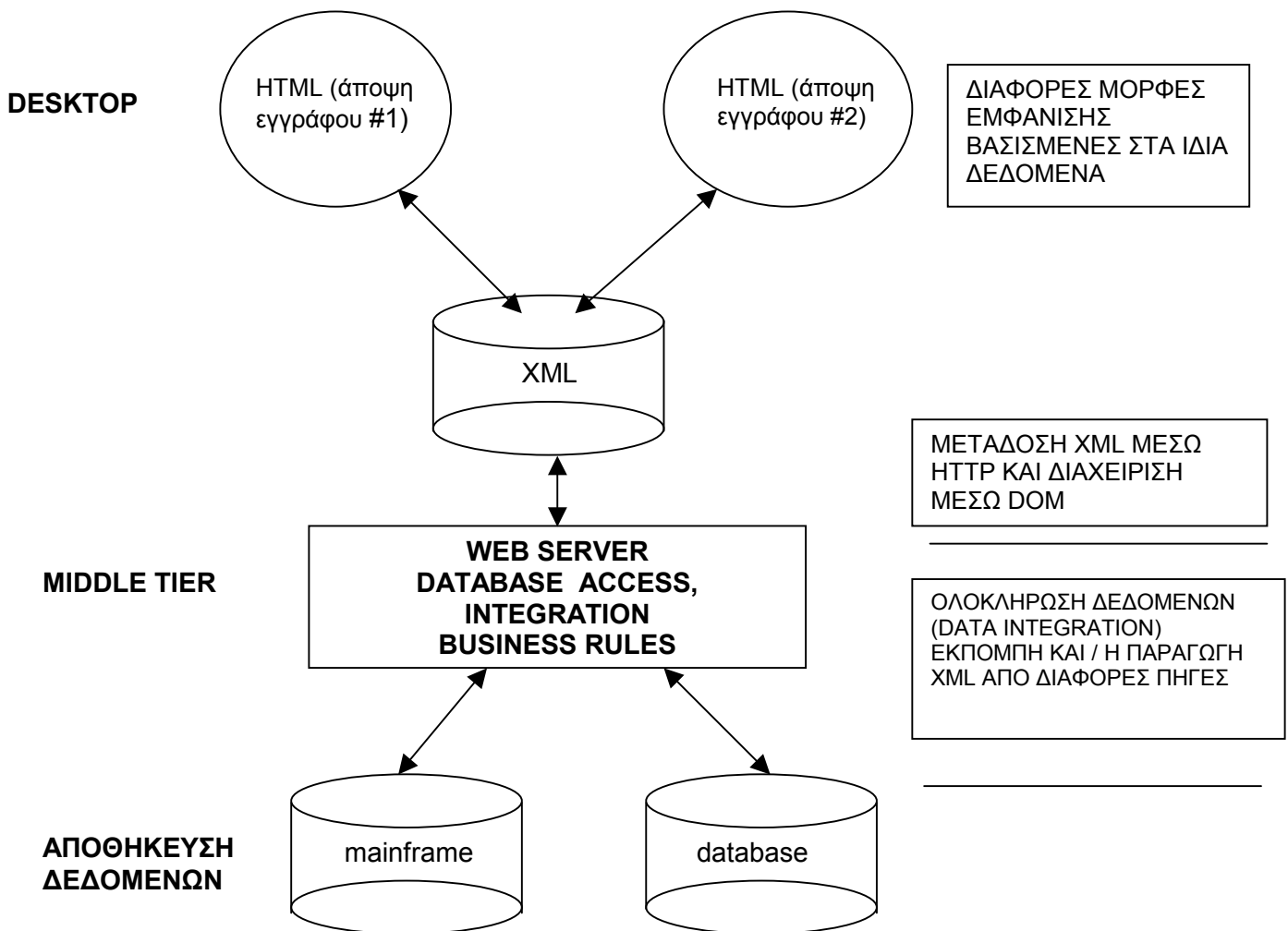
Τα business services είναι μια εφαρμογή της XML που λειτουργεί σαν agent. Παίρνει τις αιτήσεις του χρήστη, αποφασίζει ποια πληροφορία να ζητήσει και πως θα την ανακτήσει. Μετά, συνθέτει την πληροφορία έτσι ώστε

να είναι χρήσιμη στο χρήστη. Π.χ. μια αίτηση του χρήστη μπορεί να παράγει 5 XML έγγραφα. Το business services tier αναλαμβάνει να τα συνδέσει σε ένα και να τα παραδώσει στο χρήστη.

✓ **User services** (Παρουσίαση σε HTML)

Τα user services προετοιμάζουν τα δεδομένα έτσι ώστε να είναι κατανοητά από το χρήστη. Το πλεονέκτημα της XML είναι ότι μπορούμε να επιλέγουμε διάφορους τρόπους παρουσίασης χωρίς να επεμβαίνουμε στα δεδομένα. Υπάρχουν διάφοροι τρόποι υλοποίησης του τρίτου επιπέδου. Εμείς στην παρούσα εργασία επιλέξαμε τη μετατροπή των εγγράφων XML σε HTML στο server μέσω της XSL.

Όλα τα παραπάνω γίνονται πιο κατανοητά από το παρακάτω σχέδιο.



Σχέδιο 1

## Υλοποίηση εφαρμογής

Αρχικά θα παρουσιαστεί ο τρόπος με το λιγότερο δυνατό προγραμματιστικό όγκο, ώστε προχωρώντας να εξετάζουμε πιο πολύπλοκες περιπτώσεις. Η Microsoft Access μας δίνει τη δυνατότητα δημιουργίας active server pages αυτόματα, μέσω ερωτημάτων. Αυτή η μέθοδος παρουσιάζεται παρακάτω, ωστόσο δε συνιστάται γιατί κρίθηκε αναξιόπιστη και δημιουργεί κώδικα σε HTML, ενώ εμείς προτιμούμε το φίλτρο XSL όπως θα δούμε αργότερα. Επίσης, χρειαζόμαστε δύο αρχεία asp για τη δημιουργία XML εγγράφων, ενώ ένα είναι αρκετό όπως θα δούμε παρακάτω.

Ένα από τα στοιχεία που θα θέλαμε να παρακολουθούμε είναι και η λήψη παραγγελιών ανά μήνα. Για το σκοπό αυτό δημιουργούμε ένα ερώτημα στη βάση δεδομένων με το όνομα ORDERSMHNA. Αφού το σώσουμε σε μορφή ASP με το όνομα ORDERSMHNA.asp σχεδιάζουμε τη δομή του εγγράφου XML που θέλουμε να δημιουργήσουμε. Η προκειμένη περίπτωση είναι πολύ απλή. Δημιουργούμε 2 tags με το όνομα ΜΗΝΑΣ και ΠΛΗΘΟΣΜΗΝΑ. Το πρώτο, θα περιέχει τις ονομασίες των μηνών και το δεύτερο το πλήθος των παραγγελιών που έλαβε η εταιρεία. Επίσης, επειδή στην XML πρέπει πάντα να υπάρχει ένα αρχικό στοιχείο (root element) δημιουργούμε το στοιχείο SALES.

Έπειτα γράφουμε κώδικα σε VBScript για την παραγωγή του εγγράφου.

### Αρχείο Ordmonth.asp

```
<%@LANGUAGE = VBScript%>
<?xml version="1.0" encoding="ISO-8859-7"?>
' Δήλωση της XML
<Ordersmhna>
<%
    Set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open "orders"
    Set Eleggxos = Conn.Execute("select * from Ordersmhna")
    Do While Not Eleggxos.EOF
    %>
<SALES>
```



```
<ΜΗΝΑΣ><%=Elegxos("ΜΗΝΑΣ")%></ΜΗΝΑΣ>
<ΠΛΗΘΟΣΜΗΝΑ><%=Elegxos("Sum")%></ΠΛΗΘΟΣΜΗΝΑ>
</SALES>
<% Elegxos.MoveNext
Loop %>
</Ordersmhna>
```

Πολύ σημαντικό είναι να τονίσουμε ότι για να αναγνωρίζονται τα Ελληνικά, στη δήλωση της έκδοσης της XML θα πρέπει να δηλώνουμε **Ελληνική κωδικοποίηση (encoding="ISO-8859-7")**.

Στον παραπάνω κώδικα δημιουργούμε ένα Active Data Object με το όνομα Elegxos. Το αντικείμενο αυτό επικοινωνεί με τη βάση δεδομένων όπως φαίνεται και από τη γραμμή :

```
Set Elegxos = Conn.Execute("select * from Ordersmhna")
```

Έπειτα τα tags συμπληρώνονται με δεδομένα μέχρι να διαβαστούν όλες οι σειρές του ερωτήματος (γι αυτό ευθύνεται το loop στην προτελευταία σειρά).

Το αρχείο σώθηκε σαν ordmonth.asp.

Το αποτέλεσμα από την εκτέλεση αυτού του αρχείου είναι ένα έγγραφο σε XML που στον Internet Explorer εμφανίζεται σε δομημένη μορφή, ως εξής :

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<Ordersmhna>
  <SALES>
    <ΜΗΝΑΣ>ΙΑΝΟΥΑΡΙΟΣ</ΜΗΝΑΣ>
    <ΠΛΗΘΟΣΜΗΝΑ>5</ΠΛΗΘΟΣΜΗΝΑ>
  </SALES>
  <SALES>
    <ΜΗΝΑΣ>ΔΕΚΕΜΒΡΙΟΣ</ΜΗΝΑΣ>
    <ΠΛΗΘΟΣΜΗΝΑ>25</ΠΛΗΘΟΣΜΗΝΑ>
  </SALES>
</Ordersmhna>
```

Έτσι, με ένα απλό script μπορέσαμε να παράγουμε ένα έγγραφο XML. Ωστόσο, αυτή ήταν και η απλούστερη περίπτωση που θα συναντήσουμε.

Παρακάτω θα εξάγουμε δεδομένα από τη βάση μας και θα τα μετατρέψουμε σε XML με βάση ένα χαρακτηριστικό που θα δίνεται σε πραγματικό χρόνο.

Ο σκοπός του ερωτήματος είναι η ανεύρεση ενός εμβόλου το οποίο η εταιρεία διαθέτει σε stock. Για το σκοπό αυτό δημιουργούμε μια φόρμα σε HTML, τη φόρμα FINDSTOCK.html, γράφοντας τον παρακάτω κώδικα :

**Αρχείο FINDSTOCK.html**

```
<HTML>
<HEAD><b>ΕΥΡΕΣΗ ΕΜΒΟΛΩΝ ΣΕ STOCK
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html;charset=windows-1253">
<TITLE>FindStock</TITLE>
<BODY>
<FORM METHOD="GET" ACTION="FindStock.ASP">
Εισάγετε το min απαιτούμενο μήκος σε mm
<P><INPUT TYPE="Text" NAME="[ENTER min]"><P>
<P>
<hr>
Εισάγετε το maximum απαιτούμενο μήκος σε mm <P>
<INPUT TYPE="Text" NAME="[enter max]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FORM>
</BODY>
</HTML>
```

Συνήθως οι μηχανικοί της εταιρείας ψάχνουν ένα αποθηκευμένο έμβολο βάση του μήκους του. Με την παραπάνω φόρμα ο χρήστης εισάγει το minimum και το maximum μήκος εμβόλου που απαιτεί κάποια παραγγελία. Πατώντας το κουμπί εκτέλεσης του ερωτήματος, καλείται το αρχείο FINDSTOCK.asp. Το αρχείο αυτό αναλαμβάνει να βρει τα δεδομένα τα οποία θέλουμε και να τα μετατρέψει σε xml. Ο κώδικας του αρχείου είναι σε Vbscript και είναι ο παρακάτω :

**Αρχείο FINDSTOCK.asp**

```
<%@ LANGUAGE=VBSCRIPT%>
<%Response.ContentType="text/plain"%>
```

```

<%
Dim EmvName
Dim EmvQuantity
Dim DIAR

        If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
        Else
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "orders","",""
Set Session("orders_conn") = conn
End If

sql = "SELECT stocks.EmvName, stocks.EmvQuantity, Diair.Perig,
stocks.Emvnum FROM stocks INNER JOIN Diair ON stocks.DIAIR =
Diair.Diair WHERE

        (((stocks.EmvQuantity) Between " & Request.QueryString("[ENTER
min]") & " And " & Request.QueryString("[enter max]") & ")) "
If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)
End If
%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="FstockStyle.xsl"?>
<FINDSTOCK xmlns="x-schema:FSchema.xml">
<%
        Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "orders"
Set Eleg = Conn.Execute(SQL)
Do While Not Eleg.EOF
%>
<seekstock>
        <EmvName><%=Eleg("EmvName")%></EmvName>
        <EmvQuantity><%=Eleg("EmvQuantity")%></EmvQuantity>
        <DIAIR><%=Eleg("Perig")%></DIAIR>
</seekstock>

```

```
<% Eleg.MoveNext  
Loop %>  
</FINDSTOCK>
```

Αφού δηλώσουμε τη γλώσσα συγγραφής του κώδικα, ορίζουμε τα αντικείμενα τα οποία θα αναζητήσουμε :

```
Dim EmvName  
Dim EmvQuantity  
Dim DIAR
```

Έπειτα εκτελούμε ένα ερώτημα σε SQL. Το σημείο που θα πρέπει να προσέξουμε εδώ, είναι το μήκος του εμβόλου, δηλαδή το EmvQuantity. Επειδή παίρνει τιμές από τη φόρμα, το δηλώνουμε ως :

```
Between " & Request.QueryString("[ENTER min]") & " And " &  
Request.QueryString("[enter max]")
```

Εναλλακτικά βέβαια θα μπορούσαμε να εξισώσουμε το ζητούμενο μήκος με μία παράμετρο, τεχνική που θα χρησιμοποιηθεί αργότερα.

Στη συνέχεια ακολουθεί ο κώδικας της δημιουργίας της απάντησης σε xml. Το αποτέλεσμα είναι ένα έγγραφο XML σε δομημένη μορφή (αφού δεν έχουμε ορίσει ακόμη κάποιο stylesheet).

Θα πρέπει να σημειώσουμε εδώ, ότι IE 5.0, έχει σαν προεπιλογή ένα στοιχειώδες stylesheet για να προβάλλει τα XML έγγραφα δομημένα.

Το τμήμα παραγωγής ενδιαφέρεται για τις απαιτήσεις των πελατών σε διάφορα υλικά έτσι ώστε από στατιστικά στοιχεία να μπορεί να προγραμματίζει τη ροή παραγωγής. Επίσης, το τμήμα προμηθειών ενδιαφέρεται για παρόμοιου τύπου στοιχεία έτσι ώστε να μπορεί να προγραμματίζει τις παραγγελίες υλικών.

Παρακάτω, θα δείξουμε πως το πλήθος των δοχείων ανά είδος θα μετατραπεί σε XML έγγραφο με ένα διαφορετικό κώδικα από τους προηγούμενους που παρουσιάστηκαν. Με τον παρακάτω κώδικα δε χρειάζεται να σώσουμε κανένα πίνακα ή ερώτημα σε μορφή ASP όπως έγινε στο πρώτο παράδειγμα :

### **Αρχείο SumOfTanks**

```
<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<TankData>
<%
SQL = "SELECT DISTINCTROW Count(OrderDoxeio.OrderID) AS
SUMofOrderID, Doxeio.DoxeioName " &_
" FROM Doxeio " &_
"INNER JOIN (ORDERS INNER JOIN OrderDoxeio ON
ORDERS.OrderID=OrderDoxeio.OrderID)ON Doxeio.DoxeioID
=OrderDoxeio.DoxeioId" &_
" GROUP BY Doxeio.DoxeioName;"

set Conn = Server.CreateObject("ADODB.Connection")
set rs = Server.CreateObject("ADODB.Recordset")

Conn.Open("orders")
set rs = Conn.Execute(SQL)

rs.MoveFirst()
while (not rs.EOF)
%>
<Doxeia>
  <DoxeioName> <%= rs("DoxeioName") %> </DoxeioName>
  <SumOfOrderID> <%= rs("SumOfOrderID") %> </SumOfOrderID>
</Doxeia>
<%
  rs.MoveNext()
wend
rs.close()
conn.close()
%>
</TankData>
```

Το αποτέλεσμα είναι ένα έγγραφο XML της μορφής :

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```
<TankData>
```

```
<Doxeia>
```

```
<DoxeioName> Δοχείο T 150 </DoxeioName>
```

```
<SumOfOrderID> 4 </SumOfOrderID>
```

```
</Doxeia>
```

```
.....
```

```
<Doxeia>
```

```
<DoxeioName> Δοχείο T1200 </DoxeioName>
```

```
<SumOfOrderID> 1 </SumOfOrderID>
```

```
</Doxeia>
```

```
</TankData>
```

Το σημείο που θα έπρεπε να προσέξουμε εδώ, είναι ότι χρησιμοποιούμε ένα ερώτημα SQL όπως ακριβώς θα το χρησιμοποιούσαμε και σε μια βάση δεδομένων. Και σε αυτήν την περίπτωση, το έγγραφο XML που παράγεται είναι string. Ωστόσο, ένα από τα ισχυρά σημεία της XML είναι το Document Object Model (DOM) το οποίο και θα παρουσιάσουμε παρακάτω.

## Document Object Model της XML

Το Document Object Model, είναι ένα σύνολο προδιαγραφών που επεξεργάζεται από το World Wide Web Consortium. Η γλώσσα που χρησιμοποιείται είναι η IDL (Interface Definition Language – ISO 14750) . Οι κύριοι στόχοι του DOM είναι :

- ✓ Να δίνει τη δυνατότητα παραγωγής ομάδας αντικειμένων (set of objects) και διεπιδράσεων (interfaces) που θα αποδίδουν το περιεχόμενο και τη δομή εγγράφων XML 1.0, χωρίς την απώλεια σημαντικών πληροφοριών.
- ✓ Να επιτυγχάνει τα παραπάνω ανεξάρτητα από πλατφόρμες
- ✓ Να εξασφαλίζει λειτουργικότητα στη δημιουργία αντικειμένων έτσι ώστε τα XML και HTML έγγραφα να μπορούν να δημιουργηθούν από την αρχή και εξ' ολοκλήρου σε DOM

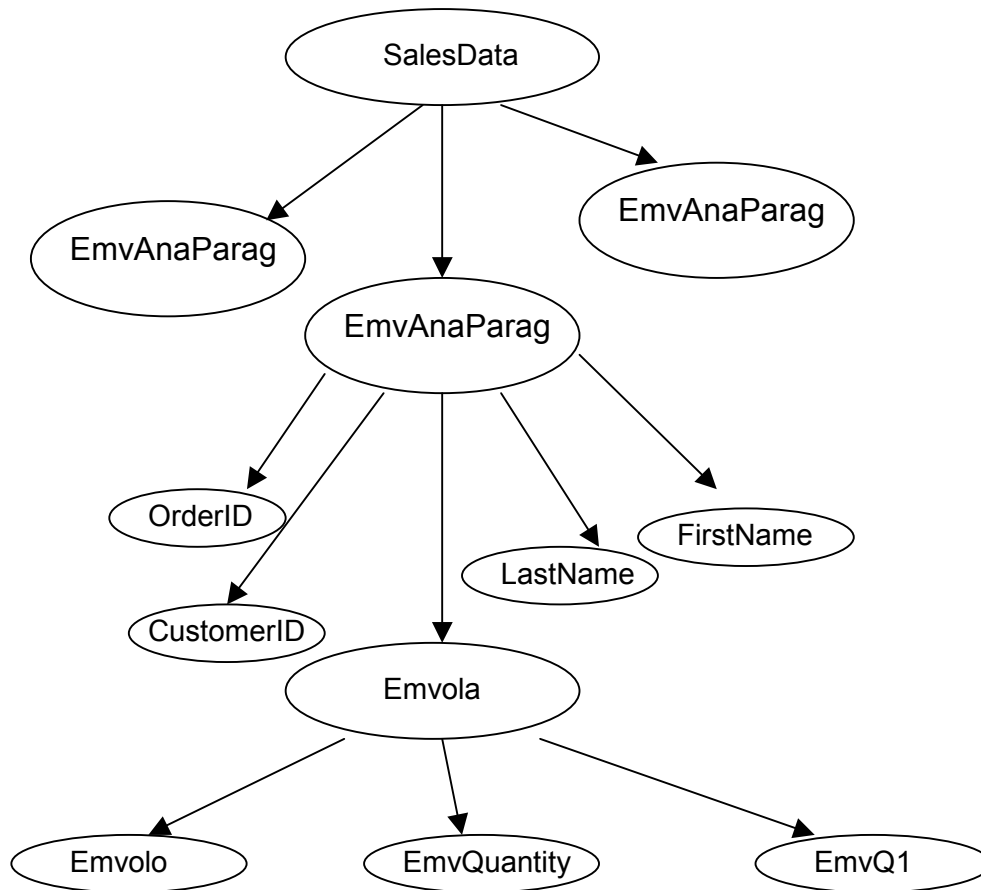
- ✓ Να είναι εφικτό να γραφτεί ένα έγγραφο από το DOM και το παραγόμενο έγγραφο να έχει την ίδια δομή με το αρχικό

Θα δημιουργήσουμε ένα έγγραφο σε μορφή DOM, κάνοντας χρήση της τεχνολογίας ASP με Vbscript. Το αποτέλεσμα του εγγράφου θα πρέπει να μας δίνει σε XML τα παρακάτω στοιχεία :

Κωδικό παραγγελίας, κωδικό πελάτη, Επώνυμο και όνομα πελάτη, και στοιχεία εμβόλου για την συγκεκριμένη παραγγελία. Ορίζουμε ένα αντικείμενο Emnola που είναι γονέας των Emnolo (περιγραφή εμβόλου), EmnQuantity (μήκος κοπής εμβόλου σε mm) Emnq1 (τεμάχια εμβόλου)

Όταν θέλουμε να δημιουργήσουμε ένα DOM είναι πολύ χρήσιμο να αναπαριστούμε σε δενδρική μορφή το έγγραφο που θέλουμε να δημιουργήσουμε, πριν γράψουμε τον κώδικα. Θα πρέπει δηλαδή να έχουμε ξεκαθαρίσει τη δομή του εγγράφου που θέλουμε να δημιουργήσουμε, τα nodes και τα childs αυτών. Τα nodes στην δενδρική μορφή των XML εγγράφων έχουν σαν «παιδιά» άλλα nodes τα οποία μπορούν και αυτά να έχουν άλλα “child nodes”. Εάν δεν έχουν, συνηθίζεται να λέγονται leaf nodes. Συγκεκριμένα, ένα αντικείμενο node μπορεί να χρησιμοποιηθεί για να αναπαραστήσει οποιοδήποτε node ενός εγγράφου και μας δίνονται ορισμένες μέθοδοι και ιδιότητες που είναι χρήσιμες για την προσέγγιση και επεξεργασία των nodes.

Το συγκεκριμένο έγγραφο θα έχει την ακόλουθη δενδρική μορφή :



Σχέδιο 2 : Δενδρική μορφή εγγράφου XML

Θα πρέπει επίσης να σημειώσουμε ότι ένα από τα πλεονεκτήματα της χρήσης του DOM είναι ότι είναι σχεδόν αδύνατο να μας δώσει έγγραφα τα οποία δεν είναι well formed.

Παρακάτω παρουσιάζουμε τον κώδικα δημιουργίας του εγγράφου σε Vbscript. Το αρχείο EmvPerOrder.asp διαβάζει δεδομένα από τη βάση δεδομένων και δημιουργεί ένα XML έγγραφο χρησιμοποιώντας το DOM. Αυτό το έγγραφο ύστερα αποστέλλεται στον client σαν text stream.

**Αρχείο EmvPerOrder.asp**

```

<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="EPOStyle.xsl" ?>
<SalesData xmlns="x-schema:EPOSchema.xml">
<%
Dim xmldoc
    
```



```
Dim root
set xmldoc = Server.CreateObject("Microsoft.XMLDOM")
' Έλεγχος για το αν το έγγραφο έχει δεδομένα. Αν έχει, να μη δημιουργηθεί
if (xmldoc.childNodes.length = 0) then
' Δημιουργία XML εγγράφου
set root = xmldoc.createElement("SalesData", "")
xmldoc.appendChild(root)
SQL = "SELECT Orders.OrderID, Customers.CustomerID, LastName,
FirstName, OrderID " & _
" FROM Customers " & _
"INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID"
SQL = SQL & " GROUP BY Customers.LastName, Customers.FirstName,
Customers.CustomerID, Orders.OrderID" & _
" HAVING (((Orders.OrderID) Like " & Request.QueryString("[enter]") & ""))
" set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open("orders")
set rs = Conn.Execute(SQL)
ocount = 0
rs.MoveFirst()
while (not rs.EOF and ocount < 100)
set onode = xmldoc.createElement("Orderssss", "")
xmldoc.documentElement.appendChild(onode)

for i = 0 to rs.Fields.Count - 1
set child = xmldoc.createElement(rs.Fields(i).Name, "")
child.text = rs.Fields(i)
onode.appendChild(child)
next
pcount = 0

iSQL = "select EmvName, OrderEmv.EmvQuantity as EmvQuantity," & _
" Emvq1 from OrderEmv " & _
" inner join Emvola on Emvola.EmvID = OrderEmv.EmvID " & _
" where OrderID = "
iSQL = iSQL & rs("OrderID")
set irs = Conn.Execute(iSQL)
irs.MoveFirst()
```

```

while (not irs.EOF and pcount < 3)
    set inode = xmldoc.createElement("element","Emvolo","")
    onode.appendChild(inode)

for i = 0 to irs.Fields.Count - 1
    set child = xmldoc.createElement("element",irs.Fields(i).Name,"")
    child.text = irs.Fields(i)
    inode.appendChild(child)
next

irs.MoveNext()
pcount = pcount + 1
wend

rs.MoveNext()
ocount = ocount + 1
wend
end if
Response.Write(xmldoc.xml)
%>
</SalesData>

```

Όπως είναι φανερό, υπάρχουν στον παραπάνω κώδικα κάποιες εντολές οι οποίες δεν είναι της Vbscript και θα τις δούμε αναλυτικότερα :

#### **DOM methods :**

*appendChild (new\_node)* : προσαρτά το αντικείμενο (node) new\_node στο τέλος της λίστας των child nodes, για κάποιο node γονέα  
*createElement (node\_type, node name)* : Δημιουργεί ένα Node χρησιμοποιώντας τις παραμέτρους που δηλώθηκαν

#### **Ιδιότητες (attributes) :**

*length* : Επιστρέφει τον αριθμό των nodes  
*xml, text* : παράγει το έγγραφο σαν stream το οποίο αποστέλλεται στον client.  
 Το αποτέλεσμα της εκτέλεσης του κώδικα είναι όπως αναμενόταν , το παρακάτω :

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<SalesData>
  <EmvAnaParag>
    <OrderID>24006</OrderID>
    <CustomerID>ΚΙΤΣΩ</CustomerID>
    <LastName>ΚΑΤΣΙΩΝΗΣ</LastName>
    <FirstName>ΔΗΜΗΤΡΙΟΣ</FirstName>
    <Emvolo>
      <EmvName>150x8</EmvName>
      <EmvQuantity>4400</EmvQuantity>
      <Emvq1>2</Emvq1>
    </Emvolo>
  </EmvAnaParag>
  .....
</SalesData>
```

Αυτό το αποτέλεσμα αποστέλλεται στον client .

## ΚΕΦΑΛΑΙΟ 3

### Προσθήκη της XSL

Έως τώρα, τα έγγραφα που παρουσιάσαμε δεν ήταν δυνατό να χρησιμοποιηθούν από το χρήστη για κάποιο σκοπό. Ένα έγγραφο XML πρέπει να τροποποιηθεί σε HTML έτσι ώστε να μπορεί να προβληθεί από τον browser, και γι αυτό το σκοπό υπάρχουν διάφοροι τρόποι. Είτε μπορεί να χρησιμοποιηθεί ένα Cascading Style Sheet (CSS) αρχείο είτε το έγγραφο XML να ενσωματωθεί σε ένα έγγραφο HTML σαν data island και με τη χρήση του Data Source Object (DSO) να είναι συμβατό με τους φυλλομετρητές.

Στην παρούσα εργασία θα προτιμήσουμε τη χρήση της eXtended Stylesheet Language (XSL) και ο βασικότερος λόγος γι' αυτήν την επιλογή είναι ότι η XSL προσφέρει πολλές δυνατότητες που οι προηγούμενοι τρόποι δεν προσφέρουν. (Το DSO υλοποιείται στο αρχείο Employee.html).

Η XSL (Extensible Stylesheet Language) είναι μια γλώσσα που δημιουργεί stylesheets τα οποία περιγράφουν πως τα δεδομένα που στέλνονται στο Web μέσω XML, θα πρέπει να παρουσιαστούν στον τελικό χρήστη.

Ένας ορισμός της XSL είναι οι παρακάτω:

« Η XSL είναι μία γλώσσα για τη δημιουργία stylesheets τα οποία περιγράφουν πως πρέπει να εμφανιστούν τα XML δεδομένα που στάλθηκαν μέσω του Web, στον χρήστη. Η XSL βασίζεται στις επεκτάσεις της DSSSL (Document Style Semantics and Specification Language) και στα πρότυπα CSS (Cascading Style Sheet)

Η XSL δίνει τη δυνατότητα στον προγραμματιστή να περιγράψει ποια πεδία δεδομένων (data fields) ενός XML εγγράφου θα παρουσιαστούν καθώς και τον τρόπο και το σημείο που θα παρουσιαστούν. Όπως όλες οι παρόμοιες γλώσσες, η XSL μπορεί να χρησιμοποιηθεί για τη δημιουργία ορισμού ενός στυλ παρουσίασης ενός εγγράφου XML ή να ξαναχρησιμοποιηθεί για πολλά XML έγγραφα.

Η XSL αποτελείται από δύο μέρη :

1. Μια γλώσσα για τη μετατροπή XML εγγράφων
2. Ένα λεξικό για τις προδιαγραφές των semantics μορφοποίησης »

[<http://whatis.com/xsl.htm>]

Οι δυνατότητες που δίνει η XSL επιτρέπουν :

- ✓ Τη μορφοποίηση των βασικών στοιχείων (source elements) σύμφωνα με αύξουσα ή φθίνουσα ταξινόμηση ή με τη θέση τους ή τη μοναδικότητά τους
- ✓ Τη δημιουργία δομών μορφοποίησης συμπεριλαμβανομένων κειμένων και γραφικών
- ✓ Τον ορισμό macros μορφοποίησης τα οποία μπορούν να ξαναχρησιμοποιηθούν
- ✓ Τη δημιουργία stylesheets ανεξάρτητα από το έγγραφο XML
- ✓ Επεκτάσιμο set από αντικείμενα (objects) μορφοποίησης

Με τη χρήση της XSL ο προγραμματιστής παύει να ενδιαφέρεται άμεσα για την εμφάνιση του εγγράφου με την οποία μπορεί να ασχοληθεί πλέον κάποιος ειδικός, π.χ. ένας γραφίστας.

Η XQL (XML Query Language) είναι ένας τρόπος να εντοπίσουμε και να φιλτράρουμε τα στοιχεία ( elements) αλλά και κείμενα σε ένα έγγραφο XML . Τα αρχεία XML χρησιμοποιούνται για τη μετάδοση δεδομένων μεταξύ των ηλεκτρονικών υπολογιστών στον παγκόσμιο ιστό. Η XQL παρέχει τη δυνατότητα εύρεσης και / ή επιλογής συγκεκριμένων δεδομένων σε ένα ή περισσότερα XML έγγραφα. Η XQL βασίζεται στο συντακτικό της XSL και έχει προταθεί σαν επέκταση της τελευταίας.

«Η XQL (XML Query Language) είναι ένας τρόπος να εντοπίζουμε και να φιλτράρουμε τα στοιχεία (elements) και κείμενα σε ένα XML έγγραφο. Η XQL παρέχει εργαλεία για ανεύρεση και επιλογή συγκεκριμένων δεδομένων ενός ή περισσότερων XML εγγράφων. Βασίζεται στο συντακτικό pattern της XSL.»

[<http://master.cpe.ku.ac.th/msit/whatis/xql.htm>]

Από τα παραπάνω, φαίνεται ότι η XSL τροποποιεί την εμφάνιση αλλά και το περιεχόμενο των εμφανιζόμενων εγγράφων (μπορούμε π.χ. να εμφανίσουμε τα δεδομένα ενός XML εγγράφου που τηρούν κάποια ανισότητα). Αυτό συμβαίνει γιατί στην έκδοση 5.0 του IE , πολλές από τις προτάσεις για την eXtended Query Language (XQL) εμπεριέχονται στην XSL.

Επίσης, αφού η εφαρμογή μας αφορά σε ένα ελεγχόμενο ενδοεπιχειρησιακό περιβάλλον, μπορούμε να είμαστε σίγουροι ότι οι όλοι οι browsers θα είναι από την έκδοση 5.0 και πάνω.

Αρχικά θα παρουσιάσουμε το απλούστερο παράδειγμα XSL. Όπως έχουμε δει έχουμε δημιουργήσει ένα αρχείο asp το οποίο σαν τελικό αποτέλεσμα έχει ένα έγγραφο XML. Κάτω από τη δήλωση της έκδοσης της XML, προσθέτουμε τη δήλωση της XSL :

```
<?xml:stylesheet type="text/xsl" href="ORDERSMHNA.xsl" ?>
```

Έχοντας ήδη δημιουργήσει το αρχείο ORDERSMHNA.xsl, μπορούμε πλέον μέσω του browser να δούμε σε κατανοητή μορφή το έγγραφο XML. Το stylesheet που δημιουργήσαμε είναι το κάτωθι :

### **Αρχείο ORDERSMHNA.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <H1>ΛΗΨΕΙΣ ΠΑΡΑΓΓΕΛΙΩΝ ΑΝΑ ΜΗΝΑ </H1>
  <TABLE BORDER="1">
    <THEAD>
      <TH>ΜΗΝΑΣ</TH>
      <TH>ΛΗΨΗ ΠΑΡΑΓΓΕΛΙΩΝ ΜΗΝΑ</TH>
    </THEAD>
    <xsl:for-each select="Ordersmhna/SALES" order-by="+ ΜΗΝΑΣ">
      <TR>
        <TD ALIGN="CENTER">
          <SPAN>
            <xsl:value-of select="ΜΗΝΑΣ"/>
          </SPAN>
        </TD>
        <TD>
          <SPAN>
            <xsl:value-of select="Π/ΛΗΘΟΣΜΗΝΑ"/>
          </SPAN>
        </TD>
      </TR>
    </xsl:for-each>
  </TABLE>
</DIV>
```

Όπως είναι φανερό, η XSL γράφεται κατά τα πρότυπα XML με προσθήκες HTML.

Στον συγκεκριμένο κώδικα, αξίζει μόνο να παρατηρήσουμε ότι με την εντολή order by επιλέξαμε την ταξινόμηση ανά μήνα, ενώ το + υποδηλώνει αύξουσα ταξινόμηση.

Οι μηχανικοί της επιχείρησης αντιμετωπίζουν πολύ συχνά το πρόβλημα άμμεσης εύρεσης τεχνικών στοιχείων κάποιων προϊόντων αλλά και υλικών. Επίσης, όταν τα προϊόντα εξελίσσονται και αλλάζουν στοιχεία είναι αρκετά δύσκολο να ενημερωθούν όλοι έγκαιρα και έγκυρα. Για το σκοπό αυτό, από την υπάρχουσα βάση δεδομένων, δημιουργούμε ένα XML έγγραφο το οποίο περιέχει μερικά από τα τεχνικά στοιχεία των δοχείων. Το αρχείο asp που χρησιμοποιούμε για τη δημιουργία του εγγράφου, είναι το παρακάτω και παρατίθεται ώστε να φαίνονται τα στοιχεία που αποτελούν το XML έγγραφο.

### **Αρχείο DoxeioTech.asp**

```
% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="DoxeioTech.xsl" ?>

<TechTank>
<%
SQL = "SELECT Doxeio.DoxeioName, Doxeio.height, Doxeio.width," &_
" Doxeio.length, Doxeio.liters " &_
" FROM Doxeio" &_
" WHERE (((Doxeio.DoxeioName)<> '--'))"
set Conn = Server.CreateObject("ADODB.Connection")
set rs = Server.CreateObject("ADODB.Recordset")
Conn.Open("orders")
set rs = Conn.Execute(SQL)
rs.MoveFirst()
while (not rs.EOF)
%>
<TankDet>
    <DoxeioName> <%= rs("DoxeioName") %> </DoxeioName>
    <height> <%= rs("height") %> </height>
    <width> <%= rs("width") %> </width>
```

```
<length> <%= rs("length") %> </length>
<liters> <%= rs("liters") %> </liters>
</TankDet>
<%
  rs.MoveNext()
wend
rs.close()
conn.close()
%>
</TechTank>
```

Ο παραπάνω κώδικας εμπεριέχει τη δήλωση :

```
<?xml:stylesheet type="text/xsl" href="DoxeioTech.xsl" ?>
```

η οποία καθορίζει πιο έγγραφο μορφοποίησης πρέπει να λάβει υπ'όψη το παραγόμενο έγγραφο της XML. Το XSL αρχείο είναι το παρακάτω :

### **Αρχείο DoxeioTech.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
  <xsl:template>
    <xsl:apply-templates/>
  </xsl:template>
  <xsl:template match="/">
    <!-- Κατασκευή HTML σελίδας -->
    <html>
      <head>
        <title>Τεχνικά στοιχεία δοχείου</title>
      </head>
      <body>
        <H1>Τεχνικά στοιχεία δοχείου</H1>
        <BLOCKQUOTE>
          <P>
            <B>Παρακάτω ακολουθούν τα τεχνικά στοιχεία των δοχείων </B>
          </P>
          <P> Τα στοιχεία <B>δεν</B> είναι πραγματικά</P>
```



```

        <BLOCKQUOTE>
            <!-- Επιλογή όλων των elements των δοχείων-->
            <xsl:apply-templates select="TechTank"/>
        </BLOCKQUOTE>
    </BLOCKQUOTE>
</body>
</html>
</xsl:template>
<xsl:template match="TankDet">
    <BR/>
    <BR/>
    <!-- Δημιουργία πίνακα -->
    <TABLE BORDER="1" CELLPADDING="10">
        <TR>
            <TD>
                <B>Τύπος δοχείου:</B>
            </TD>
            <TD>
                <xsl:apply-templates select="DoxeioName"/>
            </TD>
        </TR>
        <TR BGCOLOR="#CCCCFF"> <!-- προσθήκη χρώματος -->
            <TD>
                <B>ύψος (mm):</B>
            </TD>
            <TD>
                <xsl:apply-templates select="height"/>
            </TD>
        </TR>
        <TR BGCOLOR="#CCCCFF"> <!-- προσθήκη χρώματος -->
            <TD>
                <B>πλάτος (mm) </B>
            </TD>
            <TD>
                <xsl:apply-templates select="width"/>
            </TD>
        </TR>
    </TABLE>

```

```
<TR BGCOLOR="#CCCCFF"> <!-- προσθήκη χρώματος -->
  <TD>
    <B>μήκος (mm) </B>
  </TD>
  <TD>
    <xsl:apply-templates select="length"/>
  </TD>
</TR>
<TR BGCOLOR="#CCCCFF"> <!-- προσθήκη χρώματος -->
  <TD>
    <B>χωρητικότητα (lt) </B>
  </TD>
  <TD>
    <xsl:apply-templates select="liters"/>
  </TD>
</TR>
</TABLE>
<BR/>
</xsl:template>
<xsl:template match="DoxeioName">
  <xsl:value-of/>
</xsl:template>
<xsl:template match="liters">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
<xsl:template match="length">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
<xsl:template match="width">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
```

```
<xsl:template match="height">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
</xsl:stylesheet>
```

Από τα tags της HTML φαίνεται ότι για κάθε στοιχείο δημιουργούμε έναν πίνακα που περιλαμβάνει όλα τα υποστοιχεία του element TankDet. Ένα από τα κύρια χαρακτηριστικά της XML μπορεί να γίνει άμεσα αντιληπτό. Εάν είχαμε ορίσει διαφορετικό αρχείο μορφοποίησης, τα ίδια δεδομένα θα παρουσιάζονταν στον τελικό χρήστη με άλλη μορφή.

Στη συνέχεια θα προχωρήσουμε σε μία εφαρμογή που χρησιμοποιεί περισσότερες δυνατότητες της XSL.

Πολύ συχνά η διοίκηση της εταιρείας αλλά και οι υπεύθυνοι του οικονομικού τμήματος όπως και οι συνάδελφοί τους στις πωλήσεις, θέλουν να ενημερώνονται για τα οικονομικά στοιχεία των πελατών. Τα περισσότερα οικονομικά πακέτα της αγοράς, συνδέονται και με τον SQL server και με την Access. Έτσι, μπορούμε να θεωρήσουμε δεδομένο ότι μπορούμε να έχουμε έναν πίνακα στη βάση δεδομένων μας με οικονομικά στοιχεία. Στην παρακάτω εφαρμογή μας, σκοπός μας είναι να δώσουμε τη δυνατότητα στα στελέχη της εταιρείας, να βλέπουν έναν πίνακα με το όνομα της εταιρείας, τους τζίρους προηγούμενων ετών, τις τρέχουσες αγορές τους, την ποσοστιαία μεταβολή του τζίρου τους και αυτήν σε απόλυτο μέγεθος. Έτσι, το τμήμα πωλήσεων θα μπορεί να ελέγχει ενδεχόμενες απώλειες πελατών, νεοεισελθόντες πελάτες και απότομες διακυμάνσεις στους τζίρους. Εννοείται, ότι εύκολα μπορούν να προστεθούν στοιχεία, όπως οι πιστώσεις και οι χρεώσεις των πελατών κ.τ.λ.

Αρχικά , θα αντλήσουμε τα δεδομένα σε XML με κώδικα σε VBScript :

### **Αρχείο ΟΙΚΟΝΟΜΙΚΑ.asp**

```
<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="ΟΙΚΟΝΟΜΙΚΑ.xsl"?>
<EconomicData xmlns="x-schema:OikonomikaSchema.xml">
```

```
<%  
SQL = "SELECT Oikonomika.KLEIDI, Oikonomika.EPONYMIA, [Tziros98]-  
[Tziros97] AS change, " &_  
    " Oikonomika.POLHSEIS, (([TZIROS98]/[TZIROS97]-1)*100) AS  
Perc, Oikonomika.TZIRO97, " &_  
    " Oikonomika.TZIRO98, Oikonomika.TEL_EISPR " &_  
    " FROM Oikonomika "  
  
set Conn = Server.CreateObject("ADODB.Connection")  
set rs = Server.CreateObject("ADODB.Recordset")  
  
Conn.Open("orders")  
set rs = Conn.Execute(SQL)  
  
rs.MoveFirst()  
while (not rs.EOF)  
%>  
    <Economy>  
        <KLEIDI><%= rs("KLEIDI") %></KLEIDI>  
        <EPONYMIA><%= rs("EPONYMIA") %></EPONYMIA>  
        <change><%= rs("change") %></change>  
        <TZIRO97><%= rs("TZIRO97") %></TZIRO97>  
        <TZIRO98><%= rs("TZIRO98") %></TZIRO98>  
        <POLHSEIS><%= rs("POLHSEIS") %></POLHSEIS>  
        <Perc><%= rs("Perc") %></Perc>  
    </Economy>  
%>  
    rs.MoveNext()  
wend  
rs.close()  
conn.close()  
%>  
</EconomicData>
```

Ο παραπάνω κώδικας παρατίθεται για να επισημάνουμε μερικά λεπτά σημεία. Επειδή η XSL ταξινομεί τα δεδομένα κατά προεπιλογή σαν αφαριθμητικούς χαρακτήρες, θα πρέπει να δηλώσουμε ότι τα στοιχεία

TZIROΣ 97, TZIROΣ98, POLHSEIS, change, είναι αριθμοί (dt :dt ="number"). Αυτό γίνεται στο schema το οποίο θα παρουσιαστεί στο επόμενο κεφάλαιο.

Για να μορφοποιήσουμε το παραγόμενο έγγραφο XML δημιουργούμε το stylesheet ΟΙΚΟΝΟΜΙΚΑ.xsl.

### **Αρχείο ΟΙΚΟΝΟΜΙΚΑ.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <STYLE>
          BODY {margin:0}
          .bg {font:8pt Verdana; background-color:purple; color:white}
          H1 {font:bold 14pt Verdana; width:100%; margin-top:1em}
          .row {font:8pt Verdana; border-bottom:1px solid #CC88CC}
          .header {font:bold 9pt Verdana; cursor:hand; padding:2px; border:2px
outset gray}
          .up {background-color:#DDFFDD;}
          .down {background-color:#FFDDDD;}
        </STYLE>
      </HEAD>
      <SCRIPT>
        <xsl:comment>
          <![CDATA[
function sort(field)
{
  sortField.value = field;
  <!-- set cursor to watch here? -->
  listing.innerHTML =
source.documentElement.transformNode(stylesheet);
}
]]>
```

```
</xsl:comment>
</SCRIPT>
<SCRIPT for="window" event="onload">
  <xsl:comment>
    <![CDATA[
      stylesheet = document.XSLDocument;
      source = document.XMLDocument;
      sortField = document.XSLDocument.selectSingleNode("//@order-by");
    ]]>
  </xsl:comment>
</SCRIPT>
<BODY>
  <TABLE width="100%" cellspacing="0">
    <TR>
      <TD class="bg"/>
      <TD class="bg">
        <H1>
          ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ
        </H1>
      </TD>
    </TR>
    <TR>
      <TD class="bg" width="120" valign="top">
        <P>Πατήστε στις επικεφαλίδες των στηλών για
        αύξουσα ταξινόμηση</P>
        <P>Οι πελάτες που παρουσιάζουν αύξηση πωλήσεων
        χρωματίζονται με πράσινο και οι πτωτικοί με λευκό</P>
      </TD>
      <TD class="bg" valign="top">
        <DIV id="listing">
          <xsl:apply-templates match="EconomicData"/>
        </DIV>
      </TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
```

```

</xsl:template>
<xsl:template match="EconomicData">
  <TABLE STYLE="background-color:white">
    <THEAD>
      <TD width="80">
        <DIV class="header"
onClick="sort('KLEIDI')">ΚΩΔΙΚΟΣ</DIV>
      </TD>
      <TD width="200">
        <DIV class="header"
onClick="sort('EPONYMIA')">ΕΠΩΝΥΜΙΑ</DIV>
      </TD>
      <TD width="80">
        <DIV class="header"
onClick="sort('change')">ΜΕΤΑΒΟΛΗ</DIV>
      </TD>
      <TD width="80">
        <DIV class="header"
onClick="sort('TZIROS97')">ΤΖΙΡΟΣ1997</DIV>
      </TD>
      <TD width="80">
        <DIV class="header"
onClick="sort('TZIROS98')">ΤΖΙΡΟΣ1998</DIV>
      </TD>
      <TD width="80">
        <DIV class="header"
onClick="sort('POLHSEIS')">ΠΩΛΗΣΕΙΣ2000</DIV>
      </TD>
      <TD width="80">
        <DIV class="header" onClick="sort('Perc')">ΠΟΣΟΣΤΙΑΙΑ ΜΕΤΑΒΟΛΗ
97 -98</DIV>
      </TD>
    </THEAD>
    <xsl:for-each select="Economy" order-by="POLHSEIS">
      <TR>
        <xsl:for-each select="change">
          <xsl:if expr="this.nodeType > 0">

```

```
        <xsl:attribute name="class">up</xsl:attribute>
    </xsl:if>
</xsl:for-each>
<TD>
    <DIV class="row">
        <xsl:value-of select="KLEIDI"/>
    </DIV>
</TD>
<TD>
    <DIV class="row">
        <xsl:value-of select="EPONYMIA"/>
    </DIV>
</TD>
<TD>
    <DIV class="row" STYLE="text-align:right">
        <xsl:apply-templates select="change"/>
    </DIV>
</TD>
<TD>
    <DIV class="row" STYLE="text-align:right">
<xsl:apply-templates select="TZIROS97"/>
    </DIV>
</TD>
<TD>
    <DIV class="row" STYLE="text-align:right">
        <xsl:apply-templates select="TZIROS98"/>
    </DIV>
</TD>
<TD>
    <DIV class="row" STYLE="text-align:right">
        <xsl:apply-templates select="POLHSEIS"/>
    </DIV>
</TD>
<TD>
    <DIV class="row" STYLE="text-align:right">
        <xsl:apply-templates select="Perc"/>
    </DIV>
</TD>
```



```

        </DIV>
    </TD>
</TR>
</xsl:for-each>
</TABLE>
</xsl:template>
<xsl:template match="change|TZIROS97|TZIROS98|POLHSEIS">
<xsl:eval>formatNumber(this.nodeTypeValue , "#,###,###")</xsl:eval>
</xsl:template>
<xsl:template match="Perc">
    <xsl:eval>formatNumber(this.nodeTypeValue , "0.0")</xsl:eval>%
</xsl:template>

</xsl:stylesheet>

```

Ο παραπάνω κώδικας μορφοποιεί το παραγόμενο από τον κώδικα asp έγγραφο XML σε πίνακες. Αρχικά δηλώνουμε τις γραμματοσειρές και τα χρώματα που θα χρησιμοποιήσουμε

Έπειτα ορίζουμε μια συνάρτηση (sort) η οποία θα αναλάβει την αύξουσα ταξινόμηση των στηλών όταν ο χρήστης κάνει click στην επικεφαλίδα της κάθε στήλης, εκτός αυτής των ποσοστών. Επίσης δηλώνουμε ότι κατά το άνοιγμα του εγγράφου η συνάρτηση θα ενεργοποιηθεί , επιλέγοντας τις ιδιότητες των στοιχείων (“//@order-by»). Παρατηρούμε επίσης ότι οι συναρτήσεις είναι σε CDATA (Character Data) τμήμα, έτσι ώστε ο parser του IE 5.0 να μην τις εκλάβει σαν στοιχεία (elements) της XSL. Τα στοιχεία που χρησιμοποιούμε στο συγκεκριμένο styleshett είναι τα παρακάτω :

“ *xsl template match* = “ χρησιμοποιείται για να ορίσει μία φόρμα (template) που περιέχει τις εντολές για τη μετατροπή του XML εγγράφου για τα nodes εκείνα που ταιριάζουν με ένα συγκεκριμένο πρότυπο (pattern). Π.χ. <xsl : template match = “/”>

“ *xsl-apply templates* : “ χρησιμοποιείται μέσα σε μία φόρμα (template) για να δείξει στην xsl ότι πρέπει να ψάξει και να εφαρμόσει ένα άλλο template για ένα συγκεκριμένο node.

Π.χ. <xsl : apply-templates select =”TZIROS98”/>

“ *xsl : for each select* “ χρησιμοποιείται για να δημιουργήσει ένα loop.

“ *xsl : if expr* = “ ανάλογο του if...then, χρησιμοποιείται για να δώσει ένα διαφορετικό output εάν τηρείται κάποια συνθήκη.

“ *xsl : attribute name* = “ δημιουργεί μια νέα ιδιότητα (attribute) κόμβου (node) και την προσαρτά σε κάποιο element.

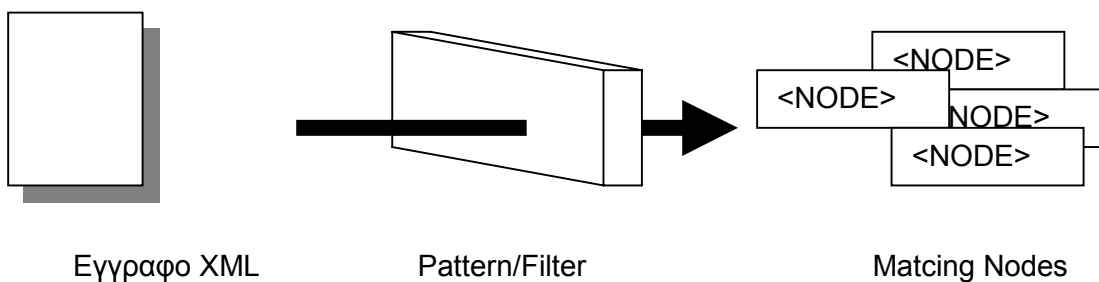
“ *xsl : value-of select* = “ χρησιμοποιείται για να αξιολογήσει ένα xsl pattern και εισάγει στο template την αξία του ταιριαστού node σαν κείμενο.

“ *xsl : eval* “ αξιολογεί μια αλφαριθμητική έκφραση και εισάγει το αποτέλεσμα στο output. Η έκφραση μπορεί να είναι μαθηματική ή λογική, script ή μια συνάρτηση της xsl.

Η παρακάτω παράγραφος εξασφαλίζει ότι όταν το μέγεθος change (TZIROS98 - TZIROS97) είναι μεγαλύτερο του 0, όταν δηλαδή ο πελάτης παρουσιάζει αύξηση τζίρου, να χρωματίζεται πράσινη η γραμμή που αναφέρεται σε αυτόν :

```
<xsl:for-each select="change">
  <xsl:if expr="this.nodeTypeValue > 0">
    <xsl:attribute name="class">up</xsl:attribute>
  < -- !Όταν το change >0 τότε η class να είναι up -- >
  </xsl:if>
</xsl:for-each>
```

Για να γίνει πιο κατανοητός ο τρόπος που η XSL μορφοποιεί τα έγγραφα, παραθέτουμε το επόμενο σχέδιο :



Σχέδιο 3

## ΚΕΦΑΛΑΙΟ 4

### XML SCHEMATA

Τα έγγραφα XML πρέπει οπωσδήποτε να είναι καλά μορφοποιημένα (well formed). Ωστόσο, αυτό δεν εξασφαλίζει ότι είναι και έγκυρα (valid).

Ένα έγγραφο είναι έγκυρο όταν ακολουθεί τις προδιαγραφές που τίθενται στο αντίστοιχο DTD (Document Type Definition). Τα DTD ορίζουν τη δομή που θα έχουν τα έγγραφα καθώς και τις ιδιότητες των στοιχείων τους. Επίσης, τα DTD μπορούν να συμπεριλαμβάνονται στο XML έγγραφο ή να είναι εξωτερικά. Ωστόσο υπάρχει μία εναλλακτική πρόταση την οποία υποστηρίζει και η Microsoft, τα **XML schemas and Data types**. Συγκεκριμένα τα XML schemata «ορίζουν τη δομή, το περιεχόμενο και την εννοιολογία (semantics) των XML εγγράφων»

[<http://www.apartmentowners.com/Standards/Presentations/SchemasForED/!tsld002.htm>]

Πιο αναλυτικά «ένα schema είναι ένα μοντέλο για την περιγραφή της δομής της πληροφορίας. Για την XML το schema περιγράφει ένα μοντέλο για μια ολόκληρη τάξη (class) εγγράφων. Το μοντέλο περιγράφει την πιθανή δόμηση tags και κειμένου σε ένα έγκυρο έγγραφο. Ένα schema μπορεί επίσης να θεωρηθεί σαν μια συμφωνία σε ένα κοινό λεξικό για μια συγκεκριμένη εφαρμογή που χρησιμοποιεί ανταλλαγή εγγράφων XML»

[<http://www.xml.com/pub/1999/07/schemas/whatis.html>]

Αυτή η τεχνολογία είναι προς το παρόν σε στάδιο πρότασης και ένα μόνο μέρος της υποστηρίζεται από τον IE 5.0. Στην παρούσα εργασία θα προτιμήσουμε αυτή τη λύση , για τους παρακάτω λόγους :

- ✓ Είναι πιο εύκολα από τα DTD και στην κατανόηση και στη χρήση
- ✓ Υποστηρίζουν τις προδιαγραφές data types
- ✓ Απλοποιούν το σχεδιασμό των εφαρμογών
- ✓ Παρέχουν καλύτερη υποστήριξη στα namespaces
- ✓ Είναι φυσική εξέλιξη των DTD
- ✓ Είναι επεκτάσιμα, έτσι ώστε στο μέλλον εύκολα θα αποκτήσουν νέα χαρακτηριστικά

Οι παραπάνω λόγοι προβάλλονται από τη Microsoft, ωστόσο και εμείς διαπιστώσαμε ότι τα schemata είναι μια σαφώς πιο κομψή, επεκτάσιμη και φιλική λύση από τα DTD.

Στο αρχείο ΟΙΚΟΝΟΜΙΚΑ .asp προσθέτουμε την παρακάτω δήλωση :

```
<EconomicData xmlns="x-schema:OikonomikaSchema.xml">
```

Το παραγόμενο αρχείο XML δηλαδή πρέπει να είναι σύμφωνο με τις προδιαγραφές που θέτουμε στο schema, το οποίο ακολουθεί :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="KLEIDI" dt:type="string" />
  <ElementType name="EPONYMIA" dt:type="string" />
  <ElementType name="change" dt:type="fixed.14.4"/>
  <ElementType name="TZIROS97" dt:type="fixed.14.4"/>
  <ElementType name="TZIROS98" dt:type="fixed.14.4"/>
  <ElementType name="POLHSEIS" dt:type="fixed.14.4"/>
  <ElementType name="Perc"/>
  <ElementType name="Economy" content="eltOnly">
    <element type="KLEIDI"/>
    <element type="EPONYMIA"/>
    <element type="change"/>
    <element type="TZIROS97"/>
    <element type="TZIROS98"/>
    <element type="POLHSEIS"/>
    <element type="Perc"/>

    <group minOccurs="1" maxOccurs="1">
<element type ="KLEIDI"/>
<element type ="EPONYMIA"/>
</group>

<group minOccurs="0" maxOccurs="1">
<element type ="change"/>
<element type ="TZIROS97"/>
<element type ="TZIROS98"/>
```

```
<element type ="POLHSEIS"/>  
<element type ="Perc"/>  
</group>  
  </ElementType>  
</Schema>
```

Αρχικά παρατηρούμε ότι και το ίδιο το schema δηλώνεται σαν XML έγγραφο. Αυτό το χαρακτηριστικό των schemata είναι ένα ακόμη πλεονέκτημά τους, το ότι δηλαδή συντάσσονται σε γλώσσα XML. Έπειτα δηλώνουμε τον τύπο του κάθε στοιχείου. Συγκεκριμένα, δηλώνουμε τα στοιχεία KLEIDI, EPONYMIA σαν string και τα υπόλοιπα σαν αριθμούς fixed. Έτσι, η XSL όταν καλείται να κάνει συγκρίσεις ή και πράξεις με τα στοιχεία ενός XML εγγράφου, ξέρει πώς να τα διαχειριστεί. Λεπτομερέστερα, το στοιχείο Element Type, χρησιμοποιείται για ορίσει τον τύπο ενός στοιχείου, η ιδιότητα content ορίζει τον τύπο των περιεχομένων του στοιχείου, η ιδιότητα dt καθορίζει τον τύπο των δεδομένων και το name δίνει το όνομα στο στοιχείο. Στη συνέχεια δηλώνουμε το root element Economy με την παρατήρηση ότι το περιεχόμενό του μπορεί να είναι μόνο άλλα elements.

Τα schemata μας δίνουν τη δυνατότητα να ορίσουμε ένα group στοιχείων για τα οποία μπορούμε να καθορίσουμε κάποιες ιδιότητες. Έτσι, ορίσαμε δύο groups με διαφορετικές ιδιότητες. Στο πρώτο, με την γραμμή

```
<group minOccurs="1" maxOccurs="1">
```

δηλώνουμε ότι τα στοιχεία που περιλαμβάνονται σε αυτήν την ομάδα είναι υποχρεωτικό να εμφανίζονται ακριβώς μία φορά σε ένα έγγραφο, ενώ στο δεύτερο, με τη δήλωση :

```
<group minOccurs="0" maxOccurs="1">
```

ορίζουμε σαν προαιρετική την εμφάνιση των συμπεριλαμβανομένων στοιχείων τα οποία μπορούν να εμφανιστούν μία φορά σε κάθε στιγμιότυπο (instance) ενός element.

Ακολουθώντας το συντακτικό της XML, πριν τελειώσει το έγγραφο, κλείνουμε όλα τα tags που έχουμε ορίσει.

## ΚΕΦΑΛΑΙΟ 5

### ΟΛΟΚΛΗΡΩΜΕΝΗ ΠΑΡΟΥΣΙΑΣΗ ΕΓΓΡΑΦΟΥ & ΕΠΕΚΤΑΣΕΩΝ ΤΟΥ

Στο παρακάτω κεφάλαιο, θα παρουσιάσουμε ολοκληρωμένα έγγραφα XML και θα επισημάνουμε κάποιες λύσεις από τα προβλήματα που συχνά συναντήθηκαν σε ηλεκτρονικές λίστες. Επίσης, θα προσθέσουμε στο intranet, μερικά ακόμη χαρακτηριστικά τα οποία θα το καταστήσουν πιο λειτουργικό.

Πολλά στελέχη της εταιρείας, επιθυμούν να ξέρουν τις εκκρεμότητες των παραγγελιών των πελατών για διάφορους λόγους, καθώς και την επιθυμητή ημερομηνία παράδοσής τους. Γι αυτό το λόγο , θα συνθέσουμε ένα έγγραφο το οποίο θα δίνει δυναμικά τα κυριότερα στοιχεία των πελατών των παραγγελιών. Αρχικά δημιουργούμε το XML έγγραφο σε συνδυασμό με μια δυναμική φόρμα. Ο ενδιαφερόμενος δηλαδή, είτε θα πληκτρολογεί την επωνυμία του πελάτη είτε θα κάνει χρήση wildcards, και συγκεκριμένα του συμβόλου «%» (ενώ στην SQL το αντίστοιχο σύμβολο είναι το “\*”).

Η φόρμα είναι σε HTML και χρησιμοποιεί τη μέθοδο get για να παίρνει τις παραμετρικές τιμές. Θα πρέπει να επισημάνουμε και την ύπαρξη της μεθόδου POST. «Η βασική τους διαφορά είναι ότι η μέθοδος GET παίρνει πληροφορίες από τη φόρμα ενώ με τη μέθοδο POST μπορούμε να ενημερώσουμε τα δεδομένα στον server.»

[<http://www.pmel.noaa.gov/toga-tao/get-vs-post.html>]

Επειδή η εφαρμογή της μεθόδου GET είναι απλούστερη, όταν δεν συντρέχει λόγος, την προτιμούμε.

#### **Αρχείο orderscustomers.html**

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html;charset=windows-1253">
<TITLE>orderscustomer</TITLE>
<BODY>
<FORM METHOD="GET" ACTION="orderscustomers.ASP">
```

```
<br>Εισάγετε την επωνυμία του πελάτη για <br>
να δείτε τις παραγγελίες που εκκρεμούν <br>
καθώς και τον επιθυμητό χρόνο παράδοσης<br>
Εάν δεν είστε σίγουροι για την επωνυμία<br>
χρησιμοποιείστε το σύμβολο % <br>
π.χ. ANY%<br><br>
<INPUT TYPE="Text" NAME="[enter]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FORM>
</BODY>
</HTML>
```

Όπως βλέπουμε, η φόρμα καλεί το αρχείο ordercustomers.asp για να πάρει τιμές.

### **Αρχείο ordercustomers.asp**

```
<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="OrderCust.xsl"?>
<OrdersC xmlns="x-schema:OrderCuSchema.xml">
<%
sql = "SELECT ORDERS.OrderID, Customers.CustomerID,
Customers.LastName, Customers.FirstName, ORDERS.OrderDate,
ORDERS.RequiredDate FROM Customers INNER JOIN ORDERS ON
Customers.customerID = ORDERS.CustomerID "

sql = sql & " GROUP BY ORDERS.OrderID, Customers.LastName,
Customers.FirstName, ORDERS.OrderDate, ORDERS.RequiredDate,
Customers.CustomerID HAVING (((Customers.LastName) Like " &
Request.QueryString("[enter]") & "%")) "

set Conn = Server.CreateObject("ADODB.Connection")
set rs = Server.CreateObject("ADODB.Recordset")

Conn.Open("orders")
set rs = Conn.Execute(SQL)
```

```
rs.MoveFirst()
while (not rs.EOF)
%>
  <OrdersCust>
    <OrderID><%= rs("OrderID") %></OrderID>
    <CustomerID><%= rs("CustomerID") %></CustomerID>
    <LastName><%= rs("LastName") %></LastName>
    <FirstName><%= rs("FirstName") %></FirstName>
    <OrderDate><%= rs("OrderDate") %></OrderDate>
    <RequiredDate><%= rs("RequiredDate") %></RequiredDate>
  </OrdersCust>
<%
  rs.MoveNext()
wend
rs.close()
conn.close()
%>
</OrdersC>
```

Η παραπάνω μέθοδος παραγωγής XML εγγράφου έχει ξαναπαρουσιαστεί στην παρούσα εργασία αλλά όχι με παραμετρικά στοιχεία. Το πρόβλημα που προέκυψε και συναντήθηκε συχνά σε ηλεκτρονικές λίστες, ήταν ότι η φόρμα δε δεχόταν τις τιμές που εισάγαμε από τη φόρμα. Η λύση ήταν η προσθήκη των κιτρινοισμένων εισαγωγικών....

Στη συνέχεια ορίσαμε το παρακάτω schema για το παραγόμενο έγγραφο xml :

### **Αρχείο OrderCuSchema.xml**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="OrderID" dt:type="i4"/>
  <ElementType name="CustomerID" dt:type="string"/>
  <ElementType name="LastName" dt:type="string"/>
  <ElementType name="FirstName" dt:type="string"/>
  <ElementType name="OrderDate" content="textOnly" dt:type="date"/>
```



```
<ElementType name="RequiredDate" content="textOnly"
dt:type="date"/>
<ElementType name="OrdersCust" content="eltOnly">
  <element type="OrderID"/>
  <element type="CustomerID"/>
  <element type="LastName"/>
  <element type="FirstName"/>
  <element type="OrderDate"/>
  <element type="RequiredDate"/>
  <group minOccurs="1" maxOccurs="1">
    <element type="OrderID"/>
    <element type="CustomerID"/>
    <element type="LastName"/>
  </group>
  <group minOccurs="0" maxOccurs="1">
    <element type="FirstName"/>
    <element type="OrderDate"/>
    <element type="RequiredDate"/>
  </group>
</ElementType>
</Schema>
```

Το τελευταίο στάδιο είναι η σύνθεση ενός ή παραπάνω XSL εγγράφου, ανάλογα με τις ανάγκες των χρηστών

### **Αρχείο OrderCust.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/W3C-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <STYLE>
          BODY {margin:0}
          .bg {font:8pt Verdana; background-color:blue; color:white}
          H1 {font:bold 14pt Verdana; width:100%; margin-top:1em}
```

```
.row {font:9pt Arial; border-bottom:1px solid #CC88CC}  
.header {font:bold 9pt Verdana; cursor:hand; padding:2px; border:2px  
outset gray}
```

```
</STYLE>  
</HEAD>  
<BODY>  
  <TABLE width="100%" cellspacing="0">  
    <TR>  
      <TD class="bg">  
        <H1>  
          ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ  
        </H1>  
      </TD>  
    </TR>  
    <TR>  
      <TD class="bg" valign="top">  
        <DIV id="listing">  
          <xsl:apply-templates match="OrdersC"/>  
        </DIV>  
      </TD>  
    </TR>  
  </TABLE>  
</BODY>  
</HTML>  
</xsl:template>  
<xsl:template match="OrdersC">  
  <TABLE STYLE="background-color:white">  
    <THEAD>  
      <TD width="40">  
        <DIV class="header">No Παραγγελίας</DIV>  
      </TD>  
      <TD width="50">  
        <DIV class="header">Κωδικός Πελάτη</DIV>  
      </TD>  
      <TD width="100">  
        <DIV class="header">Επωνυμία</DIV>
```

```
</TD>
<TD width="100">
  <DIV/>
</TD>
<TD width="80">
  <DIV class="header">Ημερ/νία Παραγγελίας</DIV>
</TD>
<TD width="80">
  <DIV class="header">Επιθυμητή Παράδοση</DIV>
</TD>
</THEAD>
<xsl:for-each select="OrdersCust" order-by="LastName">
  <TR>
    <TD>
      <DIV class="row">
        <xsl:value-of select="OrderID"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row">
        <xsl:value-of select="CustomerID"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row">
        <xsl:value-of select="LastName"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row">
        <xsl:value-of select="FirstName"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row">
        <xsl:value-of select="OrderDate"/>
      </DIV>
    </TD>
  </TR>
</xsl:for-each>
```

```

        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="RequiredDate"/>
            </DIV>
        </TD>
    </TR>
</xsl:for-each>
</TABLE>
</xsl:template>
<xsl:template
match="OrderID|CustomerID|LastName|FirstName|OrderDate|RequiredDate
"/>
</xsl:stylesheet>

```

Πολύ σημαντικό είναι, τα ερωτήματα να είναι δυνατό να υποβάλλονται έχοντας σαν παράμετρο ημερομηνίες. Για το σκοπό αυτό θα παραθέσουμε ένα ακόμη παράδειγμα στο οποίο το στέλεχος της εταιρείας μπορεί να έχει πληροφορία για τον αριθμό των σασσί του παραγγέλθηκαν σε ένα συγκεκριμένο χρονικό διάστημα. Αρχικά δημιουργούμε μια φόρμα για την εισαγωγή των στοιχείων :

### **Αρχείο SassiCount.html**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Μέτρηση Σασσί</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff text=#000000>
<BODY>
<FORM METHOD="GET" ACTION="CountSassi.ASP">
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600>

<p><FONT SIZE="4" COLOR=blue FACE="Verdana"><b>
Εισάγετε τις ημερομηνίες με την ακόλουθη μορφοποίηση :<br>

```

```
31/12/99 για 31 Δεκεμβρίου 1999 </font></P>
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600><br>
<FONT SIZE="3" COLOR="#800000" FACE="Verdana"><b>Καταμέτρηση
παραγγελιών σασσί από : <br>
<b> <INPUT TYPE="Text" NAME="[startdate]"><br><br>
έως <br>

<INPUT TYPE="Text" NAME="[enddate]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FONT></FORM>
</BODY>
</HTML>
```

Κατά τη συνήθη πλέον τακτική μας , δημιουργούμε ένα asp αρχείο το οποίο μέσω ενός sql ερωτήματος, διαχωρίζει τα πεδία απ' όπου ο κώδικας μας θα πάρει πληροφορίες για να συνθέσει το XML έγγραφο.

### **Αρχείο Count Sassi.asp**

```
<%@ LANGUAGE=VBSCRIPT%>
<%Response.ContentType="text/xml"%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="SassiSales.xsl" ?>
<SalesSassi xmlns="x-schema:SassiCSchema.xml">
<%
Dim SassilD
Dim SassiName
Dim SumOfOrderID
Dim fidate
Dim ladate
fidate=Request.QueryString("[startdate]")
ladate=Request.QueryString("[enddate]")

If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
Else
Set conn = Server.CreateObject("ADODB.Connection")
```

```
conn.open "orders", "", ""
Set Session("orders_conn") = conn
End If

sql = "SELECT DISTINCTROW Sassi.SassiID, Sassi.SassiName,
Count(ORDERS.OrderID) AS SumOfOrderID FROM Sassi INNER JOIN
(ORDERS INNER JOIN OrderSassi ON ORDERS.OrderID =
OrderSassi.OrderID) ON Sassi.SassiID = OrderSassi.SassiID WHERE
(((ORDERS.OrderDate) Between " & "#" & fidate & " # And #" & ladate & "#"
& ")) "
If cstr(Param) <> "" And cstr(Data) <> "" Then
    sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)
End If
sql = sql & " GROUP BY Sassi.SassiID, Sassi.SassiName ORDER BY
Sassi.SassiID, Count(ORDERS.OrderID) DESC "
%>
<bedates>
<fidates><%=fidate%></fidates>
<ladates><%=ladate%></ladates>
</bedates>
<%
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "orders"
Set Eleg = Conn.Execute(SQL)
Do While Not Eleg.EOF
    %>
    <SassiSales>
        <SassiID><%=Eleg("SassiID")%></SassiID>
        <SassiName><%=Eleg("SassiName")%></SassiName>
        <SumOrders><%=Eleg("SumOfOrderID")%></SumOrders>
    </SassiSales>

<% Eleg.MoveNext
Loop %>
</SalesSassi>
```

Θα πρέπει να επισημάνουμε στον αναγνώστη τη χρήση των "#" χαρακτήρων στο ερώτημα, όπως επίσης το ότι το στοιχείο SalesSassi έχει δύο υποστοιχεία (κι όχι μόνο ένα) τα οποία είναι ρίζες για άλλα στοιχεία και δε μπορούν να έχουν περιεχόμενα παρά μόνο άλλα στοιχεία, το <bedates> και το </SassiSales>. Έτσι, το αρχείο XML που δημιουργείται είναι της μορφής :

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="SassiSales.xsl" ?>
<SalesSassi xmlns="x-schema:SassiCSchema.xml">
  <bedates>
    <fidates>01/01/98</fidates>
    <ladates>01/01/00</ladates>
  </bedates>

  <SassiSales>
    <SassiID>A</SassiID>
    <SassiName>ΔιαφυγούσαΠώληση</SassiName>
    <SumOrders>6</SumOrders>
  </SassiSales>

```

---

```

<SassiSales>
  <SassiID>ΩΗ</SassiID>
  <SassiName>Πι - 3000 Kgr</SassiName>
  <SumOrders>1</SumOrders>
</SassiSales>
</SalesSassi>

```

Το Schema που ακολουθεί είναι προσαρμοσμένο στην παραπάνω δομή του εγγράφου :

### **Αρχείο SassiCSchema.xml**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="SumOfOrders" dt:type="i4"/>
  <ElementType name="SassiID" dt:type="string"/>

```

```
<ElementType name="SassiName" dt:type="string"/>
<ElementType name="fidates" content="textOnly" dt:type="date"/>
<ElementType name="ladates" content="textOnly" dt:type="date"/>
<ElementType name="bedates" content="eltOnly">
  <element type="fidates"/>
  <element type="ladates"/>
  <group minOccurs="1" maxOccurs="1">
    <element type="fidates"/>
    <element type="ladates"/>
  </group>
</ElementType>
<ElementType name="SassiSales" content="eltOnly">
  <element type="SassiID"/>
  <element type="SassiName"/>
  <element type="SumOfOrders"/>
  <group minOccurs="0" maxOccurs="1">
    <element type="SassiID"/>
    <element type="SassiName"/>
    <element type="SumOfOrders"/>
  </group>
</ElementType>
</Schema>
```

Το τελευταίο στάδιο είναι και πάλι αυτό της συγγραφής ενός xsl αρχείου :

**Αρχείο SassiSales.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:for-each select="SalesSassi/bedates">
    Πωλήσεις σασσί από : <B>
      <xsl:value-of select="fidates"/>
    </B>
    έως : <B>
      <xsl:value-of select="ladates"/>
    </B>
  </xsl:for-each>
<br/>
```



```
<TABLE BORDER="1">
  <THEAD>
    <TH>Κωδικός</TH>
    <TH>Τύπος</TH>
    <TH>Πωλήσεις</TH>
  </THEAD>
  <xsl:for-each select="SalesSassi/SassiSales" order-by="+
SumOrders">
    <TR>
      <TD STYLE="padding-left: 1em">
        <SPAN>
          <xsl:value-of select="SassiID"/>
        </SPAN>
      </TD>
      <TD STYLE="padding-left: 1em">
        <SPAN>
          <xsl:value-of select="SassiName"/>
        </SPAN>
      </TD>
      <TD STYLE="text-align:center">
        <SPAN>
          <xsl:value-of select="SumOrders"/>
        </SPAN>
      </TD>
    </TR>
  </xsl:for-each>
</TABLE>
</DIV>
```

Τα παραπάνω βήματα είναι και τα απαραίτητα για να εφαρμοστεί η αρχιτεκτονική Three Tier , βασισμένη στην τεχνολογία XML. Ωστόσο, με την αναμενόμενη εξέλιξη και των υπόλοιπων επεκτάσεων της XML, είναι σίγουρο ότι θα γίνει ακόμη πιο δυναμική και ευέλικτη τεχνολογία. Ενδεχομένως, σε βραχύ χρονικό διάστημα, κάποια από τα πρότυπα που εφαρμόσαμε παραπάνω να μην ισχύουν , με πιθανότερο αυτό της XSL, είναι όμως σχεδόν βέβαιο ότι οι βασικές αρχές της νέας αυτής τεχνολογίας και των επεκτάσεών της θα παραμείνουν ίδιες.

## Εφαρμογή EXTRANET

Όπως είδαμε παραπάνω μία εφαρμογή extranet καλύπτει την ανταλλαγή πληροφοριών με συνεργάτες, πελάτες κ.λ.π. Προσθέσαμε λοιπόν στις δυνατότητες του δικτυακού μας τόπου, μία επιλογή που δίνει την δυνατότητα στον πελάτη να βλέπει το υπόλοιπό του (χρεωστικό ή πιστωτικό), την τελευταία ημέρα που πλήρωσε κάποιον λογαριασμό, το όριο του ανοίγματος που μπορεί να έχει κ.λ.π.

Η πρόσβαση στη βάση δεδομένων από τον πελάτη, γίνεται αφού συμπληρώσει την επωνυμία της εταιρείας του σε μία φόρμα και στη συνέχεια έναν κωδικό . Εάν ο κωδικός είναι λανθασμένος δεν παίρνει απάντηση. Το πρότυπο XML μας επιτρέπει την εφαρμογή αυτής της λειτουργίας όπως και παρόμοιων, χωρίς αλλαγή στον κώδικα. Πρόσθετες ασφαλιστικές δικλείδες πρόσβασης είναι θέμα asp, διαμοιρασμού αρχείων κ.λ.π. και δε θα μας απασχολήσουν στην παρούσα εργασία.

## Πρόσθετες λειτουργίες του Intranet

Ένα Intranet επιτελεί βέβαια περισσότερες λειτουργίες από την υποβολή ερωτημάτων. Στην παρούσα εργασία επικεντρωθήκαμε στη χρήση της XML, ωστόσο θα πρέπει να προσθέσουμε και άλλες λειτουργίες έτσι ώστε να προταθεί στην υπό μελέτη εταιρεία ένα ολοκληρωμένο εσωτερικό δίκτυο. Για το σκοπό αυτό αυτό, προσθέσαμε κάποιες εφαρμογές οι οποίες είναι διαθέσιμες με freeware άδειες χρήσης στο διαδίκτυο.

Στην αρχική σελίδα προστέθηκε η επιλογή «Στατιστικά Παραπόνων / Βλαβών», μια εξαιρετικά χρήσιμη εφαρμογή XML και asp. Τα αρχεία asp είναι γραμμένα σε Vbscript. Τα αρχεία με την επέκταση inc υποδηλώνουν με αυτήν την επέκταση ότι πρέπει να εκτελεστούν στον server. Η εφαρμογή αυτή καταγράφει σε ένα αρχείο XML «ψήφους» οι οποίες προσαρμόστηκαν στις ανάγκες μας, αφού δίνεται και η δυνατότητα διαχείρισης των αρχείων χωρίς συγγραφή κώδικα. Αξιοσημείωτο είναι επίσης ότι τα αποτελέσματα παρουσιάζονται σε γραφική μορφή, με την προσθήκη κώδικα Java στο χρησιμοποιούμενο stylesheet.

Ο εμπονομαζόμενος Survey server μπορεί να βρεθεί στο URL : <http://www.siteexperts.com/tips/xml/ts04/page1.asp> και συγγραφέας του είναι ο Scott Isaacs.

Επίσης προσθέσαμε τη δυνατότητα διαγραφής και ενημέρωσης του πίνακα stocks της βάσης δεδομένων μέσω κώδικα asp. Με το αρχείο

moddata.asp μας δίνεται η δυνατότητα να ενημερώνουμε πίνακες της βάσης δεδομένων μέσω πρωτοκόλλου . Το αρχείο προσαρμόστηκε στις ανάγκες της εργασίας και μπορεί να βρεθεί στο URL :

<http://www.haneng.com/lessons/21.asp>

Προσθέσαμε επίσης τον Index Server που διατίθεται με τον IIS 4.0 σαν εσωτερικό μηχανισμό αναζήτησης , προσαρμοσμένο στις ανάγκες μας.

Παρακάτω θα παρουσιάσουμε δύο επιπλέον χαρακτηριστικά τα οποία προσθέσαμε, τα «κανάλια» και μία εφαρμογή ομαδικής εργασίας (groupware).

### **Channel Definition Format**

Το CDF (Channel Definition Format) είναι ένα μέρος του IE 4.0 και άνω για τη δημιουργία push καναλιών. Με την έννοια push, εννοείται ότι ενώ στις κλασσικές σελίδες του διαδικτύου, ο χρήστης «τραβάει» πληροφορίες, στα κανάλια οι πληροφορίες ωθούνται στον χρήστη. Η υλοποίηση της push τεχνολογίας γίνεται μέσω του CDF το οποίο ακολουθεί το XML πρότυπο.

Τα ενεργά κανάλια παραδίδουν αυτόματα στους φυλλομετρητές κανάλια (σελίδες Web) στον IE. Κάθε Push τεχνολογία χρησιμοποιεί ένα ξεχωριστό CDF αρχείο που καθορίζει τη δομή του καναλιού και περιλαμβάνει τις σελίδες, εικόνες και ActiveX controls που θα προωθηθούν στο κανάλι. Το CDF επίσης περιέχει επικεφαλίδες που περιγράφουν τα περιεχόμενα του καναλιού , το χρονοδιάγραμμα ενημέρωσης και άλλες επιλογές.

[Cnetglossary :

[ <http://coverage.cnet.com/Resources/Info/Glossary/Terms/cdf.html> ]

Εισάγαμε δύο ενεργά τέτοια κανάλια στην ιστοσελίδα μας. Το ένα από αυτά news.cdf είναι το παρακάτω :

[Αρχείο news.cdf](#)

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<CHANNEL HREF="http://eymenidhs/intranet/news.html"
LASTMOD="2000-01-28T16:17" PRECACHE="YES" LEVEL="0">
  <TITLE>Τα νέα της εταιρείας</TITLE>
  <ABSTRACT>Τα τελευταία νέα της εταιρείας μας</ABSTRACT>
  <SCHEDULE STARTDATE="2000-01-28" ENDDATE="2001-01-27"
TIMEZONE="+0200">
  <INTERVALTIME DAY="1"/>
</SCHEDULE>
```

```
<LOGTARGET HREF="http://eymenidhs/intranet/log" METHOD="POST"  
SCOPE="ALL"/>  
</CHANNEL>
```

Όπως βλέπουμε πρόκειται για ένα αρχείο XML το οποίο περιλαμβάνει σχεδιασμό ενημέρωσης του καναλιού για ένα χρόνο.

## Groupwares

Σαν “groupware ορίζονται τα προγράμματα που σαν σκοπό τους έχουν την ομαδική συνεργασία ανθρώπων που χρησιμοποιούν απομακρυσμένους υπολογιστές. Οι υπηρεσίες των groupwares μπορούν να περιλαμβάνουν διαμοιρασμό ημερολογίων, ταυτόχρονη συγγραφή κειμένων. Διαχείριση ηλεκτρονικών επιστολών , διαμοιρασμένη πρόσβαση σε βάσεις δεδομένων, συνομιλία σε πραγματικό χρόνο (chat) κ.λ.π. Τα πιο δημοφιλή παραδείγματα τέτοιων εφαρμογών είναι το Lotus Notes, Microsoft Exchange, CU-SeeMe και Microsoft NetMeeting. “

[ <http://whatis.com/groupwar.htm> ]

Η λύση που χρησιμοποιήσαμε είναι το ICQ groupware καθώς παρουσιάστηκε το πιο σταθερό από όσα δοκιμάστηκαν, και διέθετε το φιλικότερο περιβάλλον.

Επίσης οι απαιτήσεις του σε πόρους είναι πολύ μικρές. Η εφαρμογή αυτή αναπτύχθηκε ειδικά για χρήση εντός των επιχειρήσεων, σε Intranet περιβάλλον. Οι υπηρεσίες που παρέχει η δωρεάν έκδοση είναι εκπομπή και ανταλλαγή μηνυμάτων, ανταλλαγή αρχείων, διαμοιρασμός βάσεων δεδομένων μέσω ODBC, διαχείριση ηλεκτρονικών, εξαιρετο interface για το διαχειριστή του δικτύου μηνυμάτων και όλες τις υπηρεσίες που παρέχει η γνωστή και διαδεδομένη έκδοση του icq για προσωπικούς υπολογιστές. Το icq groupware μπορεί να ανταποκριθεί ικανοποιητικά σε 200 το πολύ χρήστες. Το ICQ επίσης, λειτουργεί αψογα στο Web και είναι ευρέως γνωστό.

Η δικτυακή τοποθεσία του προϊόντος βρίσκεται στη διεύθυνση :

<http://www.icq.com/groupware/>

Δείγμα του icq groupware ακολουθεί στην επόμενη σελίδα .



## ΚΕΦΑΛΑΙΟ 6

### ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως πολύ συχνά προαναφέρθηκε, η τεχνολογία XML είναι επεκτάσιμη και ευέλικτη, Αυτά τα χαρακτηριστικά μας έδωσαν τη δυνατότητα να την προσαρμόσουμε στα δεδομένα μιας επιχείρησης που η παραγωγική της διαδικασία και τα προϊόντα της είναι μοναδικά στον Ελλαδικό χώρο. Με την ίδια ευκολία η XML προσαρμόζεται στις ανάγκες οποιασδήποτε εφαρμογής. Η απόλυτη ελευθερία που δίνει στον προγραμματιστή έχει βέβαια ως αντίκρισμα το χρόνο εκμάθησης, καθώς αναμφισβήτητα η HTML είναι πιο εύκολη γλώσσα. Ωστόσο, η XML παρέχει τα κύρια χαρακτηριστικά της SGML και είναι σαφώς απλούστερη. Θα μπορούσαμε ανεπιφύλακτα να πούμε, ότι για να χρησιμοποιήσει κάποιος την XML θα πρέπει να έχει γνώσεις SQL, βάσεων δεδομένων, μίας γλώσσας προγραμματισμού, λειτουργίας του μοντέλου server-client και HTML.

Μερικές επεκτάσεις της XML δεν παρουσιάστηκαν σκόπιμα, καθώς δεν υποστηρίζονται ακόμη (Φεβρουάριος 2000) από την τρέχουσα έκδοση του Internet Explorer ( 5.0). Συγκεκριμένα η XLL (XML Linking Language) και η (XPL) XML Pointer Language είναι γλώσσες που σχεδιάστηκαν από το W3C για να εφαρμόζουν δεσμούς (links) μεταξύ XML εγγράφων και των πηγών τους (resources). Οι Xpointers παρέχουν ένα συντακτικό το οποίο επιτρέπει τον καθορισμό μεμονωμένων στοιχείων ή ακόμα και μερών τους (π.χ. το τρίτο <Εμνολο> element ή ακόμη π.χ. και τα τρία τελευταία γράμματα της επωνυμίας ενός πελάτη που έχει ένα συγκεκριμένο κωδικό ). Οι XLinks χρησιμοποιούν τους Xpointers για να καθορίσουν τον προορισμό του δεσμού.

Επίσης θα πρέπει να αναφέρουμε ότι και η XSL ενδεχομένως να αλλάξει κάποια στοιχεία στο μέλλον. Ωστόσο, το ενδιαφέρον των εταιρειών, η συνεχώς αυξανόμενη χρήση του νέου προτύπου, η ανεπάρκεια των υπάρχοντων τεχνολογιών να ανταποκριθούν στις απαιτήσεις του ηλεκτρονικού εμπορίου και η μελλοντική πλήρης υποστήριξη της XML από τους δύο σημαντικότερους browsers της αγοράς, καθιστούν βέβαιη την κυριαρχία της XML σε εφαρμογές intranet, extranet και Internet.

Ήδη όπως είδαμε, ο IE 5.0 υποστηρίζει μερικώς την XML, το ίδιο και ο Netscape Navigator 5.0. Επίσης το Microsoft Office 2000 χρησιμοποιεί XML για να γιαιτηρεί τις μορφοποιήσεις των εγγράφων Word, Excel και PowerPoint όταν αυτά μετατρέπονται σε HTML έτσι ώστε όταν τα

παραγόμενα έγγραφα όταν ανοίγονται από τις εφαρμογές από τις οποίες προήλθαν να μη χάνεται η μορφοποίησή τους.

Η αποδοχή της XML αυξάνεται αλματωδώς , αφού προσφέρει έναν ισχυρό αλλά και απλό τρόπο επικοινωνίας μεταξύ διαφορετικών βάσεων δεδομένων και συστημάτων, μέσα και έξω από κάποιον οργανισμό.

Η XML χαρακτηρήστηκε σαν το «EDI των φτωχών». Το γεγονός της ανεξαρτησίας της από πλατφόρμες, το μηδαμινό κόστος εφαρμογής της και η επεκτασιμότητά της, δικαιώνουν αυτόν τον χαρακτηρισμό.

Έτσι, καλά σχεδιασμένες εφαρμογές που λαμβάνουν υπ'όψη τους το σημαντικό εργαλείο, μπορούν να γίνουν το ανταγωνιστικό πλεονέκτημα μιας εταιρείας, θεωρώντας ότι όλα τα είδη δικτύων, είτε Intranet, Extranet και Internet θα παίξουν καταλυτικό ρόλο στις αποδόσεις των επιχειρήσεων του εγγύς μέλλοντος.

## ΠΑΡΑΡΤΗΜΑ

### *Αρχεία HTML*

#### **Default.htm**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>KLEEMANN</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
<STYLE>A {
    TEXT-DECORATION: none}
</STYLE>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
<P align=center>&nbsp;</P>
<TABLE BORDER=0 CELLPADDING=1 CELLSPACING=1
WIDTH="100%" ALIGN="Left">
<TR>
<FONT color=#004080
face=Verdana size=1>

<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Στοιχεία πωλήσεων</TD>
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Οικονομικά Στοιχεία</TD>
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Τεχνικά θέματα</TD>
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Διάφορα</TD>

</TR>
<TR>
<TD><A HREF="http://eymenidhs/intranet/EmvPerOrder.html"
TARGET="_blank">Στοιχεία εμβόλου/παραγγελία</A><BR></TD>
```



```

        <TD><A HREF="http://eymenidhs/intranet/OIKONOMIKA.asp"
TARGET="_blank">Οικονομικά στοιχεία πελατών</A></TD>
        <TD><A HREF="http://eymenidhs/intranet/DoxeioTech.asp"
TARGET="_blank">Τεχνικά στοιχεία δοχείων</A></TD>
        <TD><A HREF="http://eymenidhs/intranet/moddata.asp"
TARGET="_blank">Ενημέρωση εμβόλων σε stock</A></TD>
    </TR>
    <TR>
        <TD><A HREF="http://eymenidhs/intranet/orderscustomer.html"
TARGET="_blank">Παραγγελίες πελατών </A><BR></TD>
        <TD></TD><TD></TD>
        <TD><A HREF="http://eymenidhs/intranet/survey/voter.asp"
TARGET="_blank">Στατιστικά Παραπόνων/βλαβών</A></TD>
    </TR>
    <TR>
        <TD><A HREF="http://eymenidhs/intranet/FINDSTOCK.html"
TARGET="_blank">Εμβολα από Stock </A><BR></TD>
        <TD></TD>
    </TR>
    <TR>
        <TD><A HREF="http://eymenidhs/intranet/ordmonth.asp"
TARGET="_blank">Παραγγελίες /μήνα</A></TD>
        <TD></TD>
    </TR>
    <TR>
        <TD><A HREF="http://eymenidhs/intranet/SassiCount.html"
TARGET="_blank">Πωλήσεις σασσί</A><BR></TD>
        <TD></TD>
    </TR>
    <TR>
        <TD><A HREF="http://eymenidhs/intranet/SumOfTanks.asp"
TARGET="_blank">Πωλήσεις δοχείων</A></TD>
    </TR><TR><TD></TD></TR><TR><TD></TD></TR><TR><TD></TD>
</TR><TR><TD></TD></TR><TR><TD></TD></TR><TR></FONT>
</TR></FONT></TABLE>
<P align=center><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif"><FONT color=#004080

```

```
face=Terminal size=4></FONT></P></BODY></HTML>
```

### **DiamOrders.html**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Παραγγελίες ανά διαμέρισμα</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
```

```
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
```

```
<BODY>
<FORM METHOD="GET" ACTION="DiamOrders.ASP">
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600>
<p><FONT SIZE="4" COLOR=blue FACE="Verdana"><b>
<br>
```

```
Εισάγετε τις ημερομηνίες με την ακόλουθη μορφοποίηση :<br>
31/12/99 για 31 Δεκεμβρίου 1999 </font></P>
```

```
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600><br>
```

```
Παραγγελίες από : <br><INPUT TYPE="Text" NAME="[fdate]"><P>
έως : <br><INPUT TYPE="Text" NAME="[ldate]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
```

```
</FORM>
</BODY>
</HTML>
```

### **EIOrders.html**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Έλεγχος Παραγγελιών πελατών</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
```

```

<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
<BODY>
<FORM METHOD="GET" ACTION="EIOrders.ASP">
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600>
<p><FONT SIZE="4" COLOR=blue FACE="Verdana"><b>
<br>Εισάγετε την επωνυμία του πελάτη για να δείτε τις <br>
παραγγελίες που εκκρεμούν καθώς και τον επιθυμητό χρόνο <br>
παράδοσης. Εάν δεν είστε σίγουροι για την επωνυμία<br>
χρησιμοποιήστε το σύμβολο %, π.χ. ANY%<br>
Εισάγετε τις ημερομηνίες με την ακόλουθη μορφοποίηση :<br>
31/12/99 για 31 Δεκεμβρίου 1999 </font></P>
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600><br>
Επωνυμία Πελάτη : <br><INPUT TYPE="Text" NAME="[eponymia]"><P>
Παραγγελίες από : <br><INPUT TYPE="Text" NAME="[fdate]"><P>
έως : <br><INPUT TYPE="Text" NAME="[idate]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FORM>
</BODY>
</HTML>

```

### **EmvPerOrder.html**

```

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html;charset=windows-1253">
<TITLE>Στοιχεία εμβόλου/παραγγελία</TITLE>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
<FORM METHOD="GET" ACTION="Emvperorder.ASP">
<br><b>Εισάγετε το νούμερο παραγγελίας <br>

```

να δείτε τα στοιχεία εμβόλου της<br>

```
<br>
```

```
<INPUT TYPE="Text" NAME="[enter]"><P>
```

```
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

### **Extranet.html**

```
<HTML>
```

```
<HEAD>
```

```
<META HTTP-EQUIV="Content-Type"
```

```
CONTENT="text/html;charset=windows-1253">
```

```
<TITLE>extranet</TITLE>
```

```
<META content="text/html; charset=windows-1253" http-equiv=Content-  
Type>
```

```
<META content="MSHTML 5.00.2314.1000"
```

```
name=GENERATOR></HEAD>
```

```
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff  
text=#000000>
```

```
<BODY>
```

```
<FORM METHOD="GET" ACTION="extranet.ASP">
```

```
<HEAD2><B>ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ</b></HEAD2>
```

```
<hr ALIGN=LEFT COLOR=BLUE SIZE=1 WIDTH=400>
```

```
<B>Εισαγωγή επωνυμίας εταιρείας : <br>
```

```
<INPUT TYPE="Text" NAME="[name]"><P>
```

```
Κωδικός :<br>
```

```
<INPUT TYPE="Text" NAME="[passw]"><P>
```

```
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

**FindStock.HTML**

```
<HTML>
<HEAD><b>ΕΥΡΕΣΗ ΕΜΒΟΛΩΝ ΣΕ STOCK
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html; charset=windows-1253">
<TITLE>FindStock</TITLE>
<BODY>
<FORM METHOD="GET" ACTION="FindStock.ASP">
Εισάγετε το min απαιτούμενο μήκος σε mm
<P><INPUT TYPE="Text" NAME="[ENTER min]"><P>
<P><hr>
Εισάγετε το maximum απαιτούμενο μήκος σε mm <P>
<INPUT TYPE="Text" NAME="[enter max]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FORM>
</BODY>
</HTML>
```

**links.htm**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>KLEEMANN</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
<STYLE>A {
    TEXT-DECORATION: none
}
</STYLE>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
<TABLE BORDER=0 CELLPADDING=1 CELLSPACING=1
WIDTH="100%" ALIGN="Left">
<TR>
<FONT color=#004080
face=Verdana size=1>
```

```
<TD><FONT color=#004080 face=Terminal size=4>
<FONT
face=" Arial, Helvetica, sans-serif">Links Περιοδικών -
Εντύπων<BR>Web rings <BR></TD>
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Προμηθευτές - Ανταγωνιστές</TD>
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Τεχνικά θέματα</TD>
</TR>
<TR>
<TD><A HREF="http://www.fastlane.net/~rolland/index.htm"
TARGET="_blank">Πληροφορίες για ανελκυστήρες</A><BR></TD>
<TD><A HREF="http://www.otis.com/"
TARGET="_blank">OTIS</A></TD>
<TD><A HREF="http://users.black-
hole.com/users/DRUMWIND/wg1/HYDRO.htm"
TARGET="_blank">Ηλεκτρόλυση και Εμβολο</A></TD>
</TR>
<TR>
<TD><A
href="http://www.greenspun.com/boohoo/related.tcl?page_id=rojo"
TARGET="_blank">Κατάλογος Links </A><BR></TD>
<TD><A HREF="http://www.kone.com/flash_index.html"
TARGET="_blank">KONE</A></TD>
</TR>
<TR>
<TD>
<A HREF="http://www.greenspun.com/boohoo/related.tcl?page_id=rojo"
TARGET="_blank">Κατάλογος Links </A><BR></TD>
<TD><A HREF="http://www.serge-elev.com/" TARGET="_blank">Serge-
elevators</A></TD>
</TR>
<TR>
<TD><A HREF="http://www.geocities.com/WallStreet/1484/floor8.htm"
TARGET="_blank">Web Ring</A></TD>
<TD><A HREF="http://www.deltaelevator.com/"
TARGET="_blank">Delta Elevators</A></TD>
```

```
</TR>
<TR>
  <TD><A HREF="http://www.elevator-world.com/"
TARGET="_blank">Elevator World</A></TD>
  <TD><A HREF="http://www.us.schindler.com/ "
TARGET="_blank">Schindler</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.bosco.co.kr/bldg1.htm"
TARGET="_blank">Bosco -Korea</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.mitsubishi-elevator.com/show/show.htm"
TARGET="_blank">Mitsubishi elevators</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.doverelevator.com/"
TARGET="_blank">Dover elevators</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.elevatori.it/prof-e.html"
TARGET="_blank">G.m.v.</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.leistritzcorp.com/index.html"
TARGET="_blank">Leistritzcorp</A></TD>
</TR>
<TR>
  <TD><A HREF="http://www.elevatorconcepts.com/vendor_links.htm"
TARGET="_blank">Κατάλογος Προμηθευτών</A></TD>
</TR><TR></FONT></TR></FONT></TABLE></FONT>
</BODY></HTML>
```

**main-gr.html**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>KLEEMANN</TITLE>
```

```
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
```

```
<STYLE>A {
    TEXT-DECORATION: none
}
</STYLE>
```

```
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
```

```
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
```

```
<P align=center>&nbsp;</P>
```

```
<TABLE BORDER=0 CELLPADDING=1 CELLSPACING=1
WIDTH="100%" ALIGN="Left">
```

```
<TR>
```

```
<FONT color=#004080
```

```
face=Verdana size=1>
```

```
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Στοιχεία πωλήσεων</TD>
```

```
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Οικονομικά Στοιχεία</TD>
```

```
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Τεχνικά θέματα</TD>
```

```
<TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Διάφορα</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD><A HREF="http://eymenidhs/intranet/EmvPerOrder.html"
TARGET="_blank">Στοιχεία εμβόλου/παραγγελία</A><BR></TD>
```

```
<TD><A HREF="http://eymenidhs/intranet/OIKONOMIKA.asp"
TARGET="_blank">Οικονομικά στοιχεία πελατών</A></TD>
```

```
<TD><A HREF="http://eymenidhs/intranet/DoxeioTech.asp"
TARGET="_blank">Τεχνικά στοιχεία δοχείων</A></TD>
```

```
<TD><A HREF="http://eymenidhs/intranet/moddata.asp"
TARGET="_blank">Ενημέρωση εμβόλων σε stock</A></TD>
```



```

</TR>
<TR>
  <TD><A HREF="http://eymenidhs/intranet/orderscustomer.html"
TARGET="_blank">Παραγγελίες πελατών </A><BR></TD>
  <TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Εφαρμογή Extranet</TD>
  <TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Εργαζόμενοι</TD></TD>
  <TD><A HREF="http://eymenidhs/intranet/survey/voter.asp"
TARGET="_blank">Στατιστικά Πραπτόνων/βλαβών</A></TD>
</TR>
<TR>
  <TD><A HREF="http://eymenidhs/intranet/FINDSTOCK.html"
TARGET="_blank">Εμβολα από Stock </A><BR></TD>
  <TD><A HREF="http://eymenidhs/intranet/extranet.html"
TARGET="_blank">Χρήση κωδικού από πελάτη</A></TD>
  <TD><A HREF="http://eymenidhs/intranet/recs.html"
TARGET="_blank">Αρχείο εργαζόμενων</A></TD>
</TR>
<TR>
  <TD><A HREF="http://eymenidhs/intranet/ordmonth.asp"
TARGET="_blank">Παραγγελίες /μήνα</A></TD>
  <TD><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif">Έλεγχος αποθεμάτων</TD>
</TR>
<TR>
  <TD><A HREF="http://eymenidhs/intranet/SassiCount.html"
TARGET="_blank">Πωλήσεις σασσί</A><BR></TD>
  <TD><A HREF="http://eymenidhs/intranet/ware.asp"
TARGET="_blank">Αποθέματα</A></TD>

</TR>
<TR>

  <TD><A HREF="http://eymenidhs/intranet/SumOfTanks.asp"
TARGET="_blank">Πωλήσεις δοχείων</A></TD>
</TR>

```

```

<TR>
  <TD><A HREF="http://eymenidhs/intranet/EIOrders.html"
TARGET="_blank">Έλεγχος παραγγελιών πελατών</A></TD>
</TR>
<TR>
  <TD><A HREF="http://eymenidhs/intranet/DiamOrders.html"
TARGET="_blank">Παραγγελίες ανά πόλη</A></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD></TD>
</TR>
<TR>
<TD></TD>
</TR>
</FONT>
</TR>
</FONT>
</TABLE>

```

```

<P align=center><FONT color=#004080 face=Terminal size=4><FONT
face=" Arial, Helvetica, sans-serif"><FONT color=#004080
face=Terminal size=4></FONT></P></BODY></HTML>

```

**menu-gr.html**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>MAIN menu</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>
<STYLE>A {
  TEXT-DECORATION: none
}
</STYLE>

```

```

<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=menu-gr_files/menu2.jpeg
bgColor=#FFFFFF<BODY TEXT="#FFFFFF" LINK="#FFFFFF"
VLINK="#FFFFFF"><BODY VLINK="#FFFF00">
</BODY>
</BODY>
<P></P><BR></BR><BR></BR><BR></BR><BR></BR>
<b>Κανάλια</B>
<br><hr ALIGN=LEFT COLOR=yellow SIZE=1 WIDTH=130>
<A HREF="http://eymenidhs/intranet/links.htm"
TARGET="_blank">Χρήσιμα links στο Internet</A><BR>
<A HREF="http://eymenidhs/intranet/news.html" TARGET="_blank">Τα
νέα της εταιρείας</A>
<br>
<hr ALIGN=LEFT COLOR=yellow SIZE=1 WIDTH=130>
<A HREF="http://eymenidhs/intranet/search/query.htm"
TARGET="_blank"><font =face : Verdana size =2>Εσωτερικός μηχανισμός
αναζήτησης</font></A>
<BR>
</BODY></HTML>

```

### news.html

```

<HTML>
<head>
<title>News</title>
<style TYPE="text/css">
body      {      font          :      Verdana;
           font-weight       :      normal;
           font-style        :      normal;
           color              :      black;
           text-decoration    :      none;
           margin-left        :      .05in
           }
strong{   font              :      Verdana;
           font-weight       :      bold;

```

```

        font-style      :      normal;
        color           :      black;
        text-decoration :      none
    }
a:link {
    font      :Arial Greek;
    font-weight :      bold;
    font-style :      normal;
    text-decoration :      none;
    color     :      #CC0000}
a:active {font      :      Arial Greek;
    font-weight :      bold;
    font-style  :      normal;
    text-decoration :      none;
    color       :      #FFFFFF}
a:visited {font      :      Arial Greek;
    font-weight :      bold;
    text-decoration :      none;
    font-style  :      normal;
    color       :      #CC0000}

h1 {
    font      :      Arial Greek;
    font-weight :      bold;
    font-style :      normal;
    color     :      #000000}

h2 {
    font      :      Arial Greek;
    font-weight :      normal;
    font-style :      normal;
    color     :      black;
    text-decoration :      none;}

h3 {
    font      :      Arial Greek;
    font-weight :      normal;
    font-style :      normal;
    color     :      red;
    text-decoration :      none;}

```

```
</style>
<script LANGUAGE="VBScript">
sub TransitionSeq()
    MainImage.filters.item(0).Apply()
    MainImage.style.visibility= "visible"
    MainImage.filters.item(0).play()
    NYTitle.filters.item(0).Apply()
    NYTitle.filters.item(0).Transition=3
    NYTitle.style.visibility= "visible"
    NYTitle.filters.item(0).play()

    MenuOptions.filters.item(0).Apply()
    MenuOptions.filters.item(0).Transition=8
    MenuOptions.style.visibility= "visible"
    MenuOptions.filters.item(0).play()

    WScreen.filters.item(0).Apply()
    WScreen.filters.item(0).Transition=1
    WScreen.style.visibility= "visible"
    WScreen.filters.item(0).play()
End Sub
Sub HideAll()
    Features1.style.visibility= "hidden"
    CurNews.style.visibility= "hidden"
    AddScreen.style.visibility= "hidden"
    HomeScreen.style.visibility= "hidden"
    Welcome.style.visibility= "hidden"
    AddDesk.style.visibility= "hidden"
    Gold.style.visibility= "hidden"
End Sub
Sub ShowFeatures()
    HideAll()
    Features1.style.visibility= "visible"
End Sub
Sub ShowGold()
    HideAll()
    Gold.style.visibility= "visible"
```

```
End Sub
Sub ShowCurNews()
    HideAll()
    CurNews.style.visibility= "visible"
End Sub
Sub ShowAdd()
    HideAll()
    AddScreen.style.visibility= "visible"
End Sub
Sub ShowHome()
    HideAll()
    HomeScreen.style.visibility= "visible"
End Sub
Sub ShowWelcome()
    HideAll()
    Welcome.style.visibility= "visible"
End Sub
Sub ShowDeskMenu()
    HideAll()
    AddDesk.style.visibility= "visible"
End Sub
</script>
</head>
<BODY      background=main-gr_files/logo2.jpeg      bgColor=#ffffff
text=#000000>
    <body id="myB" bgcolor="#FFFFFF"><br>
    </div>
    <b>
    <FONT COLOR="#0000FF" FACE="Courier">Τελευταία ενημέρωση:
    27/1/2000</FONT></FONT> </b></div>
    <b>
    <div ID="marquee" STYLE="position: absolute; top: 145; left: 100;
width:285; visibility: visible">
    <marquee DIRECTION="UP" WIDTH="230" BEHAVIOR="SCROLL"
SCROLLAMOUNT="1" SCROLLDELAY="4" ALIGN="MIDDLE"
HEIGHT="250" VSPACE="15" LOOP="INFINITE">
        <font FACE="Verdana" COLOR="#000000" SIZE="2">
```

```
<UL>
  <A HREF="http://eymenidhs/intranet/DoxeioTech.asp">Νέα τεχνικά
στοιχεία δοχείων
  </A> <p> <UL><LI>
  Παρακαλούμε ελέγξτε τα νέα στοιχεία & απευθυνθείται για παρατηρήσεις
στην Τεχνική Διεύθυνση</UL>
  <HR>
  <A HREF="http://www.us.schindler.com" TARGET="_blank">Νέα
Τεχνολογία ανελκυστήρων από την Schindler</A>
  </A> <p> <UL><LI>
  Συζήτηση για τις τεχνολογικές εξελίξεις στον χώρο, στο συμβούλιο της
17/02/00
  </UL><HR><br><br><br><br></ul>
<!-- /Marquee news -->
</marquee>
</body>
</html>
```

### **orderscustomer.HTML**

```
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type"
CONTENT="text/html;charset=windows-1253">
  <TITLE>orderscustomer</TITLE>
  <BODY>
  <FORM METHOD="GET" ACTION="orderscustomers.ASP">
  <br>Εισάγετε την επωνυμία του πελάτη για <br>
να δείτε τις παραγγελίες που εκκρεμούν <br>
καθώς και τον επιθυμητό χρόνο παράδοσης<br>
Εάν δεν είστε σίγουροι για την επωνυμία<br>
χρησιμοποιήστε το σύμβολο % <br>
π.χ. ANY%<br><br>
  <INPUT TYPE="Text" NAME="[enter]"><P>
  <INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
  </FORM>
</BODY></HTML>
```

**Recs.html**

```
<HTML>
<HEAD>
<TITLE>ΕΡΓΑΖΟΜΕΝΟΙ</TITLE>
<style type="text/css">
  BODY {font-family:Arial,sans-serif; font-size:12px; font-weight:normal}
  .intro {font-family:Tahoma,Verdana,Arial,sans-serif; font-size:14px; font-
weight:bold}
</style>
</HEAD>
<BODY>
<SPAN CLASS=intro>Ευρετήριο εργαζόμενων</SPAN><P>

<XML ID="dsoEMPLOYEES" SRC="employee.xml"
  onrowsdelete="deleteAction()"
  onrowsinserted="insertAction()"
  oncellchange="cellchangeAction(">
</XML>

Κωδικός: <INPUT ID="Code" TYPE="TEXT"
DATASRC="#dsoEMPLOYEES" DATAFLD="CODE" SIZE=5><P>
  Ονοματεπώνυμο: <INPUT ID="TITLE" TYPE="TEXT"
DATASRC="#dsoEMPLOYEES" DATAFLD="TITLE" SIZE=30><P>
  Ειδικότητα: <INPUT ID="TITLE" TYPE="TEXT"
DATASRC="#dsoEMPLOYEES" DATAFLD="CATEGORY"
SIZE=30><P><P>
  Ημερομηνία Πρόσληψης: <SPAN ID="Release"
DATASRC="#dsoEMPLOYEES" DATAFLD="PRDATE"></SPAN><P>
  Μισθός : <SPAN ID="Sales" DATASRC="#dsoEMPLOYEES"
DATAFLD="WAGE"></SPAN><P>
  Τμήμα : <SPAN ID="Sales" DATASRC="#dsoEMPLOYEES"
DATAFLD="TMHMA"></SPAN><P>

<button onclick="dsoEMPLOYEES.recordset.moveFirst()" TITLE="First
Record" id="cmdFirst">&nbsp;&|&lt;&nbsp;&nbsp;</button>
```



```
<button onclick="if (! dsoEMPLOYEES.recordset.EOF)
dsoEMPLOYEES.recordset.movePrevious()" TITLE="Previous
Record">&nbsp;&lt;&nbsp;&nbsp;</button>
<button onclick="if (! dsoEMPLOYEES.recordset.EOF)
dsoEMPLOYEES.recordset.moveNext()" TITLE="Next
Record">&nbsp;&gt;&nbsp;&nbsp;</button>
<button onclick="dsoEMPLOYEES.recordset.moveLast()" TITLE="Last
Record">&nbsp;&gt;&nbsp;&nbsp;</button>&nbsp;&nbsp;
<SCRIPT LANGUAGE="JScript">
function cellchangeAction() {
  strMesg = 'The following elements are bound to the DSO:\n';
  for (i = 0; i < event.boundElements.length; i++)
    strMesg += 'ID=' + event.boundElements[i].id + ' ';
  strMesg += '\n\nThe name of the field you changed is: ' + event.dataFld;
  alert(strMesg);
}
</SCRIPT>
</BODY>
</HTML>
```

### **SassiCount.HTML**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Μέτρηση Σασσί</TITLE>
<META content="text/html; charset=windows-1253" http-equiv=Content-
Type>

<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY background=main-gr_files/logo2.jpeg bgColor=#ffffff
text=#000000>
<BODY>
<FORM METHOD="GET" ACTION="CountSassi.ASP">
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600>
<p><FONT SIZE="4" COLOR=blue FACE="Verdana"><b>
Εισάγετε τις ημερομηνίες με την ακόλουθη μορφοποίηση :<br>
31/12/99 για 31 Δεκεμβρίου 1999 </font></P>
```

```
<hr ALIGN=LEFT COLOR=darkblue SIZE=2 WIDTH=600><br>
<FONT SIZE="3" COLOR="#800000"
FACE="Verdana"><b>Καταμέτρηση παραγγελιών σασσί από : <br>
<b> <INPUT TYPE="Text" NAME="[startdate]"><br><br>
έως <br>
<INPUT TYPE="Text" NAME="[enddate]"><P>
<INPUT TYPE="Submit" VALUE="Εκτέλεση ερωτήματος">
</FONT></FORM>
</BODY>
</HTML>
```

### *Αρχεία ASP*

#### **CountSassi.asp**

```
<%@ LANGUAGE=VBSCRIPT%>
<%Response.ContentType="text/xml"%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="SassiSales.xsl" ?>
<SalesSassi xmlns="x-schema:SassiCSchema.xml">
<%
Dim SassilD
Dim SassiName
Dim SumOfOrderID
Dim fdate
Dim ldate

fdate=Request.QueryString("[startdate]")
ldate=Request.QueryString("[enddate]")
If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
Else
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "orders", "", ""
Set Session("orders_conn") = conn
End If
```

```
sql = "SELECT DISTINCTROW Sassi.SassiID, Sassi.SassiName,
Count(ORDERS.OrderID) AS SumOfOrderID FROM Sassi INNER JOIN
(ORDERS INNER JOIN OrderSassi ON ORDERS.OrderID =
OrderSassi.OrderID) ON Sassi.SassiID = OrderSassi.SassiID WHERE
(((ORDERS.OrderDate) Between " & "#" & fdate & " # And #" & ldate & "#"
& ") " & ") " "
```

```
If cstr(Param) <> "" And cstr(Data) <> "" Then
```

```
sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)
```

```
End If
```

```
sql = sql & " GROUP BY Sassi.SassiID, Sassi.SassiName ORDER BY
Sassi.SassiID, Count(ORDERS.OrderID) DESC " "
```

```
%>
```

```
<bedates>
```

```
<fidates><%=fdate%></fidates>
```

```
<ladates><%=ladate%></ladates>
```

```
</bedates>
```

```
<%
```

```
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
Conn.Open "orders"
```

```
Set Eleg = Conn.Execute(SQL)
```

```
Do While Not Eleg.EOF
```

```
%>
```

```
<SassiSales>
```

```
<SassiID><%=Eleg("SassiID")%></SassiID>
```

```
<SassiName><%=Eleg("SassiName")%></SassiName>
```

```
<SumOrders><%=Eleg("SumOfOrderID")%></SumOrders>
```

```
</SassiSales>
```

```
<% Eleg.MoveNext
```

```
Loop %>
```

```
</SalesSassi>
```

### **DiamOrders.asp**

```
<%@ LANGUAGE=VBSCRIPT%>
```

```
<%Response.ContentType="text/xml"%>
```

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```
<?xml:stylesheet type="text/xsl" href="DiamSales.xsl"?>
<SalesDiam>
<%
Dim fidate
Dim ladate
fidate=Request.QueryString("[fdate]")
ladate=Request.QueryString("[ldate]")

    If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
    Else
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.open "orders", "", ""
    Set Session("orders_conn") = conn
End If
    sql = "SELECT DISTINCTROW Customers.POLH,
Count(ORDERS.OrderID) AS SumOfOrderID FROM Customers INNER
JOIN ORDERS ON Customers.customerID = ORDERS.CustomerID
WHERE (((ORDERS.OrderDate) Between " & "#" & fidate & " # And #" &
ladate & "#" & ")) "
    If cstr(Param) <> "" And cstr(Data) <> "" Then
    sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)
    End If
    sql = sql & " GROUP BY Customers.POLH ORDER BY
Customers.POLH "
%>
<bedates>
<fidates><%=fidate%></fidates>
<ladates><%=ladate%></ladates>
</bedates>
<%
    Set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open "orders"
    Set Eleg = Conn.Execute(SQL)
    Do While Not Eleg.EOF
%>
<DiamSales>
```

```
<POLH><%=Eleg("POLH")%></POLH>
<SumOrders><%=Eleg("SumOfOrderID")%></SumOrders>
</DiamSales>
<% Eleg.MoveNext
Loop %>
</SalesDiam>
```

### **DoxeioTech.asp**

```
<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="DoxeioTech.xsl" ?>
<TechTank>
<%
SQL = "SELECT Doxeio.DoxeioName, Doxeio.height, Doxeio.width," &_
" Doxeio.length, Doxeio.liters " &_
" FROM Doxeio" &_
" WHERE (((Doxeio.DoxeioName)<> '--'))"

set Conn = Server.CreateObject("ADODB.Connection")
set rs = Server.CreateObject("ADODB.Recordset")

Conn.Open("orders")
set rs = Conn.Execute(SQL)

rs.MoveFirst()
while (not rs.EOF)
%>
<TankDet>
<DoxeioName> <%= rs("DoxeioName") %> </DoxeioName>
<height> <%= rs("height") %> </height>
<width> <%= rs("width") %> </width>
<length> <%= rs("length") %> </length>
<liters> <%= rs("liters") %> </liters>
</TankDet>
<%
rs.MoveNext()
wend
```

```
rs.close()
conn.close()
%>
</TechTank>
```

### **EIOrders.asp**

```
<%@ LANGUAGE=VBSCRIPT%>
<%Response.ContentType="text/xml"%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="EIStyle.xsl" ?>
<SalesEI xmlns="x-schema:EISchema.xml">
<%
Dim OrderID
Dim LastName
Dim EIOrderID
Dim fdate
Dim ldate
Dim POLH
fdate=Request.QueryString("[fdate]")
ldate=Request.QueryString("[ldate]")
    If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
    Else
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.open "orders","",""
        Set Session("orders_conn") = conn
    End If
    sql = "SELECT Customers.LastName, Customers.FirstName,
Customers.POLH, ORDERS.OrderDate, Count(ORDERS.OrderID) AS
EIOrderID" &_
        " FROM Customers INNER JOIN ORDERS ON Customers.customerID =
ORDERS.CustomerID " &_
        " GROUP BY Customers.LastName, Customers.FirstName,
Customers.POLH, ORDERS.OrderDate " &_
        " HAVING (((Customers.LastName) " &_
```

```
" Like "" & Request.QueryString("[eponymia]") & "" ) " &_  
" AND ((ORDERS.OrderDate) Between " & "#" & fdate & " # And #" &  
ldate & "#" & ") ) "  
    If cstr(Param) <> "" And cstr(Data) <> "" Then  
        sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)  
    End If  
%>  
<bedates>  
<fidates><%=fdate%></fidates>  
<ladataes><%=ladate%></ladataes>  
</bedates>  
  
<%  
    Set Conn = Server.CreateObject("ADODB.Connection")  
    Conn.Open "orders"  
    Set Eleg = Conn.Execute(SQL)  
    Do While Not Eleg.EOF  
%>  
    <EISales>  
        <LastName><%=Eleg("LastName")%></LastName>  
        <FirstName><%=Eleg("FirstName")%></FirstName>  
        <POLH><%=Eleg("POLH")%></POLH>  
        <EIOrderID><%=Eleg("EIOrderID")%></EIOrderID>  
    </EISales>  
<% Eleg.MoveNext  
Loop %>  
</SalesEI>
```

### **Emvperorder.asp**

```
<% Response.ContentType = "text/xml" %>  
<?xml version="1.0" encoding="ISO-8859-7"?>  
<?xml:stylesheet type="text/xsl" href="EPOStyle.xsl" ?>  
<SalesData xmlns="x-schema:EPOSchema.xml">  
<%  
Dim xmldoc  
Dim root  
Dim CustomerID
```

```

Dim OrderID
Dim LastName
Dim FirstName
Dim EmvID
Dim EmvName
Dim EmvQuantity
Dim Emvq1
set xmldoc = Server.CreateObject("Microsoft.XMLDOM")
' Check to see if a document has data. If it does, don't build it

if (xmldoc.childNodes.length = 0) then
' Build the XML document
set root = xmldoc.createElement("SalesData", "")
xmldoc.appendChild(root)
SQL = "SELECT Customers.CustomerID, LastName, FirstName,
OrderID " & _
" FROM Customers " & _
" INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID"
SQL = SQL & " GROUP BY Customers.LastName,
Customers.FirstName, Customers.CustomerID, Orders.OrderID" & _
" HAVING (((Orders.OrderID) Like "" & Request.QueryString("[enter]") &
"")) "
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open("orders")
set rs = Conn.Execute(SQL)
ocount = 0
rs.MoveFirst()
while (not rs.EOF and ocount < 100)
set onode = xmldoc.createElement("EmvAnaParag", "")
xmldoc.documentElement.appendChild(onode)
for i = 0 to rs.Fields.Count - 1
set child = xmldoc.createElement(rs.Fields(i).Name, "")
child.text = rs.Fields(i)
onode.appendChild(child)
next
pcount = 0
iSQL = "select EmvName, OrderEmv.EmvQuantity as EmvQuantity," & _

```



```
" Emvq1 from OrderEmv " & _
" inner join Emvola on Emvola.EmvID = OrderEmv.EmvID " & _
" where OrderID = "
iSQL = iSQL & rs("OrderID")
set irs = Conn.Execute(iSQL)
irs.MoveFirst()
while (not irs.EOF and pcount < 5)
set inode = xmldoc.createElement("Emvolo", "")
onode.appendChild(inode)
for i = 0 to irs.Fields.Count - 1
set child = xmldoc.createElement(irs.Fields(i).Name, "")
child.text = irs.Fields(i)
inode.appendChild(child)
next
irs.MoveNext()
pcount = pcount + 1
wend
rs.MoveNext()
ocount = ocount + 1
wend
end if
Response.Write(xmldoc.xml)
%>
</SalesData>
```

### **FindStock.asp**

```
<%Response.ContentType="text/xml"%>
<%
Dim EmvName
Dim EmvQuantity
Dim DIAR
If IsObject(Session("orders_conn")) Then
Set conn = Session("orders_conn")
Else
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "orders", "", ""
Set Session("orders_conn") = conn
```

```
End If
```

```
sql = "SELECT stocks.EmvName, stocks.EmvQuantity, Diair.Perig,
stocks.Emvnum FROM stocks INNER JOIN Diair ON stocks.DIAIR =
Diair.Diair WHERE (((stocks.EmvQuantity) Between " &
Request.QueryString("[ENTER min]") & " And " &
Request.QueryString("[enter max]") & ")) "
```

```
    If cstr(Param) <> "" And cstr(Data) <> "" Then
```

```
        sql = sql & " And [" & cstr(Param) & "] = " & cstr(Data)
```

```
    End If
```

```
%>
```

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```
<?xml:stylesheet type="text/xsl" href="FstockStyle.xsl"?>
```

```
<FINDSTOCK xmlns="x-schema:FSchema.xml">
```

```
<%
```

```
    Set Conn = Server.CreateObject("ADODB.Connection")
```

```
    Conn.Open "orders"
```

```
    Set Eleg = Conn.Execute(SQL)
```

```
    Do While Not Eleg.EOF
```

```
%>
```

```
    <seekstock>
```

```
        <EmvName><%=Eleg("EmvName")%></EmvName>
```

```
        <EmvQuantity><%=Eleg("EmvQuantity")%></EmvQuantity>
```

```
        <DIAIR><%=Eleg("Perig")%></DIAIR>
```

```
    </seekstock>
```

```
<% Eleg.MoveNext
```

```
Loop %>
```

```
</FINDSTOCK>
```

### **moddata.asp**

```
<HTML>
```

```
<BODY>
```

```
<B>Μεταβολή στοιχείων εμβόλων από stock</B>
```

```
<%
```

```
Actionvar=Request.QueryString("actionvar")
```

```
Set conn = server.createobject("adodb.connection")
```

```
DSNtemp="DRIVER={Microsoft Access Driver (*.mdb)}; "
```





```

        rs.Close
    END IF
    ELSEIF Request.Form("flag")="2" THEN
        SQLstmt = "UPDATE stocks SET "
        SQLstmt = SQLstmt & "EmvNum=" &
        TRIM(Request.Form("NoEmbolou")) & ", "
        SQLstmt = SQLstmt & "EmvName=" & TRIM(Request.Form("Item")) & ", "
        SQLstmt = SQLstmt & "EmvQuantity=" & TRIM(Request.Form("Length")) &
        ", "
        SQLstmt = SQLstmt & "DIAIR=" & TRIM(Request.Form("DIAIR"))
        SQLstmt = SQLstmt & " WHERE ID=" &
        TRIM(Request.Form("Recordid"))
        %>
        <HR>
        SQL statement: <%=SQLstmt%>
        <HR>
        <%
        Conn.Execute (SQLstmt)
        Response.Write "Operation Complete<br><A
        HREF=""moddata.asp"">Home</A>"
        END IF

```

```

'If the QueryString has the Actionvar = delete then delete theType
'and generate an "Operation Complete" page
ELSEIF Actionvar="delete" THEN
    SQLstmt = "DELETE * FROM stocks WHERE ID=" &
    TRIM(Request.QueryString("Recid"))
    %>
    <HR>
    SQL statement: <%=SQLstmt%>
    <HR>
    <%
    conn.execute(SQLstmt)
    Response.Write "Operation Complete<BR><A
    HREF=""moddata.asp"">Home</A>"

```

'If the QueryString Actionvar isn't set to anything, generate the list ofTypes

ELSE

SQLstmt = "SELECT \* FROM stocks Order by Emvnum"

%>

<HR>

SQL statement: <%=SQLstmt%>

<HR>

<%

Set rs = conn.Execute(SQLstmt)

%>

<TABLE BORDER=0>

<TR>

<TD>&nbsp;</TD>

<TD ALIGN=RIGHT><B>Τύπος</B></TD>

<TD>&nbsp;</TD>

<TD><CENTER><B>Μήκος</B></CENTER></TD>

<TD>&nbsp;</TD>

<TD><CENTER><B>NoΕμβόλου</B></CENTER></TD>

<TD>&nbsp;</TD>

<TD><CENTER><B>Διαιρούμενο</B></CENTER></TD>

<TD COLSPAN=2><CENTER><B>Modify</B></CENTER></TD>

<%

x=0

DO WHILE NOT rs.EOF

  x=x+1

  Recid = rs("ID")

  Item = rs("EmvName")

  NoEmbolou = rs("EmvNum")

  Length = rs("EmvQuantity")

  DIAIR = rs("DIAIR")

  %>

<TR>

  <TD ALIGN=LEFT><%=x%></TD>

<TD ALIGN=RIGHT><%=Item%></TD>

  <TD>&nbsp;</TD>

```
<TD ALIGN=RIGHT><%=Length%></TD>
<TD>&nbsp;</TD>
<TD ALIGN=RIGHT><%=NoEmbolou%></TD>
<TD>&nbsp;</TD>
<TD ALIGN=RIGHT><%=DIAIR%></TD>
<TD>&nbsp;</TD>
<TD><CENTER><A
HREF="moddata.asp?Actionvar=delete&Recid=<%=Recid
%>">Διαγραφή</A></CENTER></TD>
<TD><A HREF="moddata.asp?Actionvar=update&Recid=<%=Recid
%>"></A></TD>
<%
rs.MoveNext
LOOP
RS.Close
Response.Write "</TABLE>"
Response.Write "<HR><A HREF='moddata.asp?actionvar=add'>Νέα
Εγγραφή</A>"
END IF

conn.Close
Set conn = nothing
Set SQLstmt = nothing
%>
</BODY>
</HTML>
```

### **OIKONOMIKA.ASP**

```
<% Response.ContentType = "text/xml" %>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="OIKONOMIKA.xsl"?>
<EconomicData xmlns="x-schema:OikonomikaSchema.xml">
<%
SQL = "SELECT Oikonomika.KLEIDI, Oikonomika.EPONYMIA,
[Tziros98]-[Tziros97] AS change, " &_
" Oikonomika.POLHSEIS, (([TZIROS98]/[TZIROS97]-1)*100) AS Perc,
Oikonomika.TZIROS97, " &_
```

```
" Oikonomika.TZIRO98, Oikonomika.TEL_EISPR " &_
" FROM Oikonomika "
set Conn = Server.CreateObject("ADODB.Connection")
set rs = Server.CreateObject("ADODB.Recordset")

Conn.Open("orders")
set rs = Conn.Execute(SQL)

rs.MoveFirst()
while (not rs.EOF)
%>
  <Economy>
    <KLEIDI><%= rs("KLEIDI") %></KLEIDI>
    <EPONYMIA><%= rs("EPONYMIA") %></EPONYMIA>
    <change><%= rs("change") %></change>
    <TZIRO97><%= rs("TZIRO97") %></TZIRO97>
    <TZIRO98><%= rs("TZIRO98") %></TZIRO98>
    <POLHSEIS><%= rs("POLHSEIS") %></POLHSEIS>
    <Perc><%= rs("Perc") %></Perc>
  </Economy>

<%
  rs.MoveNext()
wend
rs.close()
conn.close()
%>
</EconomicData>
```

**orderscustomers.asp**

```
<% Response.ContentType = "text/xml" %>
<%
sql = "SELECT  ORDErs.OrderID, Customers.CustomerID,
Customers.LastName, Customers.FirstName, ORDErs.OrderDate,
ORDErs.RequiredDate FROM Customers INNER JOIN ORDErs ON
Customers.customerID = ORDErs.CustomerID "
```



```
sql = sql & " GROUP BY Orders.OrderID, Customers.LastName,  
Customers.FirstName, ORDERs.OrderDate, Orders.RequiredDate,  
Customers.CustomerID HAVING (((Customers.LastName) Like "" &  
Request.QueryString("[enter]") & "")) "
```

```
%>
```

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```
<?xml:stylesheet type="text/xsl" href="OrderCust.xsl"?>
```

```
<OrdersC xmlns="x-schema:OrderCuSchema.xml">
```

```
<%
```

```
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
Conn.Open "orders"
```

```
Set rs = Conn.Execute(SQL)
```

```
Do While Not rs.EOF
```

```
%>
```

```
<OrdersCust>
```

```
<OrderID><%= rs("OrderID") %></OrderID>
```

```
<CustomerID><%= rs("CustomerID")%></CustomerID>
```

```
<LastName><%= rs("LastName") %></LastName>
```

```
<FirstName><%= rs("FirstName") %></FirstName>
```

```
<OrderDate><%= rs("OrderDate") %></OrderDate>
```

```
<RequiredDate><%= rs("RequiredDate")%></RequiredDate>
```

```
</OrdersCust>
```

```
<%
```

```
rs.MoveNext()
```

```
Loop%>
```

```
</OrdersC>
```

### **ORDERMHNA.asp**

```
<HTML>
```

```
<HEAD>
```

```
<META HTTP-EQUIV="Content-Type"
```

```
CONTENT="text/html;charset=windows-1253">
```

```
<TITLE>ORDERSMHNA</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<%
```

```

Param = Request.QueryString("Param")
Data = Request.QueryString("Data")
%>
<%
If IsObject(Session("_conn")) Then
    Set conn = Session("_conn")
Else
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.open ""","",""
    Set Session("_conn") = conn
End If
%>
<%
    sql = "SELECT DISTINCTROW mhnes1.MHNAΣ,
If(IsNull([OrdersMonth].[SUMofOrderID]),0,[OrdersMonth].[SUMofOrderID])
AS Sum FROM mhnes1 INNER JOIN OrdersMonth ON mhnes1.NO =
OrdersMonth.Month "
    If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " WHERE [" & cstr(Param) & "] = " & cstr(Data)
    End If
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.Open sql, conn, 3, 3
%>
<TABLE BORDER=1 BGCOLOR=#ffffff CELLSPACING=0><FONT
FACE="Arial"
COLOR=#000000><CAPTION><B>ORDERSMHNA</B></CAPTION>
<THEAD>
<TR>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>MHNAΣ</FONT></TH>
<TH BGCOLOR=#c0c0c0 BORDERCOLOR=#000000 ><FONT SIZE=2
FACE="Arial" COLOR=#000000>Sum</FONT></TH>
</TR>
</THEAD>
<TBODY>
<%
On Error Resume Next

```

```
rs.MoveFirst
do while Not rs.eof
  %>
  <TR VALIGN=TOP>
    <TD BORDERCOLOR=#c0c0c0 ><FONT SIZE=2 FACE="Arial"
COLOR=#000000><%=Server.HTMLEncode(rs.Fields("ΜΗΝΑΣ").Value)%>
<BR></FONT></TD>
    <TD BORDERCOLOR=#c0c0c0 ALIGN=RIGHT><FONT SIZE=2
FACE="Arial"
COLOR=#000000><%=Server.HTMLEncode(rs.Fields("Sum").Value)%><B
R></FONT></TD>
  </TR>
  <%
rs.MoveNext
loop%>
</TBODY>
<TFOOT></TFOOT>
</TABLE>
</BODY>
</HTML>
```

### **ordmonth.asp**

```
<%@LANGUAGE = VBScript%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="ORDERSMHNA.xsl" ?>
<Ordersmhna>
  <%
    Set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open "orders"
    Set Elegxos = Conn.Execute("select * from Ordersmhna")
    Do While Not Elegxos.EOF
      %>
      <SALES>
        <ΜΗΝΑΣ><%=Elegxos("ΜΗΝΑΣ")%></ΜΗΝΑΣ>
        <ΠΛΗΘΟΣΜΗΝΑΣ><%=Elegxos("Sum")%></ΠΛΗΘΟΣΜΗΝΑΣ>
      </SALES>
    <% Elegxos.MoveNext
```

```
Loop %>  
</Ordersmhna>
```

### **SumOfTanks.asp**

```
<% Response.ContentType = "text/xml" %>  
<?xml version="1.0" encoding="ISO-8859-7"?>  
<?xml:stylesheet type="text/xsl" href="TankSalStyle.xsl"?>  
<TankData>  
<%  
    SQL = "SELECT DISTINCTROW Count(OrderDoxeio.OrderID) AS  
SUMofOrderID, Doxeio.DoxeioName " &  
    " FROM Doxeio " &  
    "INNER JOIN(ORDERS INNER JOIN OrderDoxeio ON  
ORDERS.OrderID=OrderDoxeio.OrderID)ON Doxeio.DoxeioID  
=OrderDoxeio.DoxeioId" &  
    " GROUP BY Doxeio.DoxeioName;"  
  
    set Conn = Server.CreateObject("ADODB.Connection")  
    set rs = Server.CreateObject("ADODB.Recordset")  
  
    Conn.Open("orders")  
    set rs = Conn.Execute(SQL)  
  
    rs.MoveFirst()  
    while (not rs.EOF)  
    %>  
        <Doxeia>  
            <DoxeioName> <%= rs("DoxeioName") %> </DoxeioName>  
            <SumOfOrderID> <%= rs("SumOfOrderID") %> </SumOfOrderID>  
        </Doxeia>  
    <%  
        rs.MoveNext()  
    wend  
    rs.close()  
    conn.close()  
%></TankData>
```

**Ware.asp**

```
<% Response.ContentType = "text/xml" %>
<%
    sql = " SELECT WAREHOUSE.CODE, WAREHOUSE.CATEGORY,
WAREHOUSE.LAST_ORDER, WAREHOUSE.YLIKO,
WAREHOUSE.STOCK, WAREHOUSE.DES_STOCK, [stock]-[des_stock]
AS DIAFORA FROM WAREHOUSE "
%>
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml:stylesheet type="text/xsl" href="WARE.xsl" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"

xmlns:dt="urn:schemas-microsoft-com:datatypes">
<STOCKMAT>
<%
Set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open "orders"
    Set rs = Conn.Execute(SQL)
    Do While Not rs.EOF
%>
    <MATSTOCK>
        <ΚΩΔΙΚΟΣ><%= rs("CODE") %></ΚΩΔΙΚΟΣ>
        <ΚΑΤΗΓΟΡΙΑ><%= rs("CATEGORY")%></ΚΑΤΗΓΟΡΙΑ>
        <ΤΕΛΕΥΤΑΙΑ_ΠΑΡΑΓΓΕΛΙΑ><%= rs("LAST_ORDER")
%></ΤΕΛΕΥΤΑΙΑ_ΠΑΡΑΓΓΕΛΙΑ>
        <ΥΛΙΚΟ><%= rs("ΥΛΙΚΟ") %></ΥΛΙΚΟ>
        <ΑΠΟΘΕΜΑ dt:type="fixed.14.4"><%= rs("STOCK")
%></ΑΠΟΘΕΜΑ>
        <ΕΠΙΘΥΜΗΤΟ_ΑΠΟΘΕΜΑ dt:type="fixed.14.4"><%=
rs("DES_STOCK")%></ΕΠΙΘΥΜΗΤΟ_ΑΠΟΘΕΜΑ>
        <DIAFORA><%= rs("DIAFORA") %></DIAFORA>
    </MATSTOCK>

<%
    rs.MoveNext()
Loop%>
```

```
</STOCKMAT>  
</Schema>
```

### *Αρχεία CDF*

#### **Links.cdf**

```
<?XML VERSION="1.0" ENCODING="windows-1253"?>  
  
<CHANNEL HREF="http://eymenidhs/intranet/links.htm"  
LASTMOD="2000-01-28T00:44" PRECACHE="YES" LEVEL="0">  
  <TITLE>Usefull links</TITLE>  
  <ABSTRACT>Θέματα υδραυλικού ανελκυστήρα στο  
Internet</ABSTRACT>  
  <LOGTARGET HREF="http://eymenidhs/intranet/default.htm"  
METHOD="POST" SCOPE="ALL">  
    <PURGETIME HOUR="12"/>  
  </LOGTARGET>  
</CHANNEL>
```

#### ***news.cdf***

```
<?xml version="1.0" encoding="ISO-8859-7"?>  
<CHANNEL HREF="http://eymenidhs/intranet/news.html"  
LASTMOD="2000-01-28T16:17" PRECACHE="YES" LEVEL="0">  
  <TITLE>Τα νέα της εταιρείας</TITLE>  
  <ABSTRACT>Τα τελευταία νέα της εταιρείας μας</ABSTRACT>  
  <SCHEDULE STARTDATE="2000-01-28" ENDDATE="2001-01-27"  
TIMEZONE="+0200">  
    <INTERVALTIME DAY="1"/>  
  </SCHEDULE>  
  <LOGTARGET HREF="http://eymenidhs/intranet/log"  
METHOD="POST" SCOPE="ALL"/>  
</CHANNEL>
```

## Αρχεία XML (shemata)

### EISchema.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="LastName" dt:type="string"/>
  <ElementType name="FirstName" dt:type="string"/>
  <ElementType name="POLH" dt:type="string"/>
  <ElementType name="EIOrderID" dt:type="fixed.14.4"/>
  <ElementType name="ladates" content="textOnly" dt:type="date"/>
  <ElementType name="fidates" content="textOnly" dt:type="date"/>
  <ElementType name="bedates" content="eltOnly">
    <element type="fidates"/>
    <element type="ladates"/>
    <group minOccurs="1" maxOccurs="1">
      <element type="fidates"/>
      <element type="ladates"/>
    </group>
  </ElementType>
  <ElementType name="EISales" content="eltOnly">
    <element type="LastName"/>
    <element type="FirstName"/>
    <element type="POLH"/>
    <element type="EIOrderID"/>
    <element type="bedates"/>
    <group minOccurs="1" maxOccurs="1">
      <element type="LastName"/>
    </group>
    <group minOccurs="0" maxOccurs="1">
      <element type="FirstName"/>
      <element type="POLH"/>
      <element type="EIOrderID"/>
    </group>
  </ElementType>
</Schema>
```

**EPOSchema.xml**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">

  <ElementType name="CustomerID" content="textOnly" dt:type="string" />
  <ElementType name="LastName" content="textOnly" dt:type="string" />
  <ElementType name="FirstName" content="textOnly" dt:type="string" />
  <ElementType name="OrderID" content="textOnly" dt:type="i1" />
  <ElementType name="EmvName" content="textOnly" />
  <ElementType name="EmvQuantity" content="textOnly" />
  <ElementType name="Emvq1" content="textOnly" />
  <ElementType name="Emvolo" content="eltOnly">

    <element type="EmvName" />
    <element type="EmvQuantity" />
    <element type="Emvq1" />
    </ElementType >
    <ElementType name="EmvAnaParag" content="eltOnly">
      <element type="CustomerID"/>
        <element type="LastName"/>
        <element type="FirstName"/>
        <element type="OrderID"/>
        <element type="Emvolo"/>
    </ElementType>
  <ElementType name="SalesData" content="eltOnly">
    <element type = "EmvAnaParag" minOccurs="0" maxOccurs="*" />
  </ElementType>
</Schema>
```

**ExtranetSchema.xml**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="EPONYMIA" dt:type="string"/>
  <ElementType name="KODIKOS" dt:type="string"/>
  <ElementType name="POLHSEIS" dt:type="fixed.14.4"/>
```



```

<ElementType name="ΥΠΟΛΟΙΠΟ" dt:type="fixed.14.4"/>
<ElementType name="LAST" content="textOnly" dt:type="date"/>
<ElementType name="ORIO" dt:type="fixed.14.4"/>
<ElementType name="PISTOTIKA" dt:type="fixed.14.4"/>
<ElementType name="Extra" content="eltOnly">
  <element type="EPONYMIA"/>
  <element type="KODIKOS"/>
  <element type="POLHSEIS"/>
  <element type="ΥΠΟΛΟΙΠΟ"/>
  <element type="LAST"/>
  <element type="ORIO"/>
  <element type="PISTOTIKA"/>
  <group minOccurs="1" maxOccurs="1">
    <element type="EPONYMIA"/>
    <element type="KODIKOS"/>
  </group>
  <group minOccurs="0" maxOccurs="1">
    <element type="POLHSEIS"/>
    <element type="ΥΠΟΛΟΙΠΟ"/>
    <element type="LAST"/>
    <element type="ORIO"/>
    <element type="PISTOTIKA"/>
  </group>
</ElementType>
</Schema>

```

### **Fschema.xml**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="EmvName" dt:type="string"/>
  <ElementType name="EmvQuantity" dt:type="i4"/>
  <ElementType name="DIAIR" dt:type="string"/>
  <ElementType name="seekstock" content="eltOnly">
    <element type="EmvName"/>

```

```

<element type="EmvQuantity"/>
<element type="DIAIR"/>
<group minOccurs="0" maxOccurs="*">
  <element type="EmvName"/>
  <element type="EmvQuantity"/>
  <element type="DIAIR"/>
</group>
</ElementType>
<ElementType name="FINDSTOCK " content="eltOnly">
  <element type="seekstock" minOccurs="0" maxOccurs="*" />
</ElementType>
</Schema>

```

### **OikonomikaSchema.xml**

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="KLEIDI" dt:type="string" />
  <ElementType name="EPONYMIA" dt:type="string" />
  <ElementType name="change" dt:type="fixed.14.4"/>
  <ElementType name="TZIROS97" dt:type="fixed.14.4"/>
  <ElementType name="TZIROS98" dt:type="fixed.14.4"/>
  <ElementType name="POLHSEIS" dt:type="fixed.14.4"/>
  <ElementType name="Perc"/>

  <ElementType name="Economy" content="eltOnly">
    <element type="KLEIDI"/>
    <element type="EPONYMIA"/>
    <element type="change"/>
    <element type="TZIROS97"/>
    <element type="TZIROS98"/>
    <element type="POLHSEIS"/>
    <element type="Perc"/>

  <group minOccurs="1" maxOccurs="1">
<element type ="KLEIDI"/>

```

```
<element type="EPONYMIA"/>
</group>
```

```
<group minOccurs="0" maxOccurs="1">
  <element type="change"/>
  <element type="TZIROS97"/>
  <element type="TZIROS98"/>
  <element type="POLHSEIS"/>
  <element type="Perc"/>
</group>
```

```
</ElementType>
</Schema>
```

### **OrderCuSchema.xml**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="OrderID" dt:type="i4"/>
  <ElementType name="CustomerID" dt:type="string"/>
  <ElementType name="LastName" dt:type="string"/>
  <ElementType name="FirstName" dt:type="string"/>
  <ElementType name="OrderDate" content="textOnly"
dt:type="date"/>
  <ElementType name="RequiredDate" content="textOnly"
dt:type="date"/>
  <ElementType name="OrdersCust" content="eltOnly">
    <element type="OrderID"/>
    <element type="CustomerID"/>
    <element type="LastName"/>
    <element type="FirstName"/>
    <element type="OrderDate"/>
    <element type="RequiredDate"/>
    <group minOccurs="1" maxOccurs="1">
      <element type="OrderID"/>
      <element type="CustomerID"/>
```

```

        <element type="LastName"/>
    </group>
    <group minOccurs="0" maxOccurs="1">
        <element type="FirstName"/>
        <element type="OrderDate"/>
        <element type="RequiredDate"/>
    </group>
</ElementType>
</Schema>

```

### **SassiCSchema.xml**

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
    <ElementType name="SumOfOrders" dt:type="i4"/>
    <ElementType name="SassiID" dt:type="string"/>
    <ElementType name="SassiName" dt:type="string"/>
    <ElementType name="fidates" content="textOnly" dt:type="date"/>
    <ElementType name="ladates" content="textOnly" dt:type="date"/>
    <ElementType name="bedates" content="eltOnly">
        <element type="fidates"/>
        <element type="ladates"/>
    <group minOccurs="1" maxOccurs="1">
        <element type="fidates"/>
        <element type="ladates"/>
    </group>
</ElementType>
<ElementType name="SassiSales" content="eltOnly">
    <element type="SassiID"/>
    <element type="SassiName"/>
    <element type="SumOfOrders"/>
    <group minOccurs="0" maxOccurs="1">
        <element type="SassiID"/>
        <element type="SassiName"/>
        <element type="SumOfOrders"/>
    </group>

```

```
</ElementType>
</Schema>
```

## *Αρχεία XSL*

### **DiamSales.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:for-each select="SalesDiam/bedates">
    Πωλήσεις σαασσί από : <B>
      <xsl:value-of select="fidates"/>
    </B>
    έως : <B>
      <xsl:value-of select="ladates"/>
    </B>
  <HR></HR>
  </xsl:for-each>
  <br/>
  <TABLE BORDER="1">
    <THEAD>
      <TH>ΠΟΛΗ</TH>
      <TH>Παραγγελίες</TH>
    </THEAD>
    <xsl:for-each select="SalesDiam/DiamSales" order-by="+ POLH">
      <TR>
        <TD STYLE="padding-left:1em">
          <SPAN>
            <xsl:value-of select="POLH"/>
          </SPAN>
        </TD>
        <TD STYLE="text-align:center">
          <SPAN>
            <xsl:value-of select="SumOrders"/>
          </SPAN>
        </TD>
      </TR>
    </xsl:for-each>
  </TABLE>
```

```
</TD>
</TR>
</xsl:for-each>
</TABLE>
</DIV>
```

### **DoxeioTech.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
  <xsl:template>
    <xsl:apply-templates/>
  </xsl:template>
  <xsl:template match="/">
    <!-- Κατασκευή HTML σελίδας -->
    <html>
      <head>
        <title>Τεχνικά στοιχεία δοχείου</title>
      </head>
      <body>
        <H1>Τεχνικά στοιχεία δοχείου</H1>
        <BLOCKQUOTE>
          <P>
            <B>Παρακάτω ακολουθούν τα τεχνικά στοιχεία των
δοχείων </B>
          </P>
          <P> Τα στοιχεία <B>δεν</B> είναι πραγματικά</P>
        <BLOCKQUOTE>
          <!-- Επιλογή όλων των elements των δοχείων-->
          <xsl:apply-templates select="TechTank"/>
        </BLOCKQUOTE>
      </BLOCKQUOTE>
    </body>
  </html>
</xsl:template>
<xsl:template match="TankDet">
  <BR/>
```

```
<BR/>
<!-- Δημιουργία πίνακα -->
<TABLE BORDER="1" CELLPADDING="10">
  <TR>
    <TD>
      <B>Τύπος δοχείου:</B>
    </TD>
    <TD>
      <xsl:apply-templates select="DoxeioName"/>
    </TD>
  </TR>
  <TR BGCOLOR="#CCCCCCFF">
    <!-- προσθήκη χρώματος -->
    <TD>
      <B>ύψος (mm):</B>
    </TD>
    <TD>
      <xsl:apply-templates select="height"/>
    </TD>
  </TR>
  <TR BGCOLOR="#CCCCCCFF">
    <!-- προσθήκη χρώματος -->
    <TD>
      <B>πλάτος (mm) </B>
    </TD>
    <TD>
      <xsl:apply-templates select="width"/>
    </TD>
  </TR>
  <TR BGCOLOR="#CCCCCCFF">
    <!-- προσθήκη χρώματος -->
    <TD>
      <B>μήκος (mm) </B>
    </TD>
    <TD>
      <xsl:apply-templates select="length"/>
    </TD>
  </TR>
</TABLE>
```

```
</TR>
<TR BGCOLOR="#CCCCCCFF">
  <!-- προσθήκη χρώματος -->
  <TD>
    <B>χωρητικότητα (lt) </B>
  </TD>
  <TD>
    <xsl:apply-templates select="liters"/>
  </TD>
</TR>
</TABLE>
<BR/>
</xsl:template>
<xsl:template match="DoxeioName">
  <xsl:value-of/>
</xsl:template>
<xsl:template match="liters">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
<xsl:template match="length">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
<xsl:template match="width">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
<xsl:template match="height">
  <FONT FACE="courier new">
    <xsl:value-of/>
  </FONT>
</xsl:template>
</xsl:stylesheet>
```



**EIStyle.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/W3D-xsl">

  <xsl:for-each select="SalesEI/bedates">
    Παραγγελίες από : <B>
      <xsl:value-of select="fidates"/>
    </B>

    έως : <B>
      <xsl:value-of select="ladates"/>
    </B><HR></HR>
  </xsl:for-each>
  <br/>
  <TABLE BORDER="1">

    <xsl:for-each select="SalesEI/EISales" >
      <TR>
        <TD STYLE="padding-left: 1em">
          <SPAN>
            <xsl:value-of select="LastName"/>
          </SPAN>
        </TD>
        <TD STYLE="padding-right: 1em">
          <SPAN>
            <xsl:value-of select="FirstName"/>
          </SPAN>
        </TD>
        <TD STYLE="text-align:center">
          <SPAN>
            <xsl:value-of select="EIOrderID"/>
          </SPAN>
        </TD>
      </TR>
    </xsl:for-each>
  </TABLE>
```

&lt;/DIV&gt;

**EPOStyle.xsl**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:for-each select="SalesData/SalesData/EmvAnaParag">
    <B> Στοιχεία Εμβόλου για την παραγγελία : <xsl:value-of
select="OrderID"/>
    </B>
    <HR/>
    <B>Πελάτης : <xsl:value-of select="LastName"/>
    </B>
    <B>
      <xsl:value-of select="FirstName"/>
    </B>
  </xsl:for-each>
  <br/>
  <TABLE BORDER="1">
    <THEAD>
      <TH>Τύπος</TH>
      <TH>Μήκος</TH>
      <TH>Ποσότητα</TH>
    </THEAD>
    <xsl:for-each select="SalesData/SalesData/EmvAnaParag/Emvolo">
      <TR>
        <TD STYLE="padding-left: 1em">
          <SPAN>
            <xsl:value-of select="EmvName"/>
          </SPAN>
        </TD>
        <TD STYLE="padding-left: 1em">
          <SPAN>
            <xsl:value-of select="EmvQuantity"/>
          </SPAN>
        </TD>
      </TR>
    </xsl:for-each>
  </TABLE>

```

```

    <TD STYLE="text-align:center">
      <SPAN>
        <xsl:value-of select="Emvq1"/>
      </SPAN>
    </TD>
  </TR>
</xsl:for-each>
</TABLE>
</DIV>

```

**ExtraStyle.xls**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <STYLE>
          BODY {margin:0}
          .bg {font:8pt Verdana; background-color:GREEN; color:white}
          H1 {font:bold 14pt Verdana; width:100%; margin-top:1em}
          .row {font:9pt Arial; border-bottom:1px solid #CC88CC}
          .header {font:bold 9pt Verdana; cursor:hand; padding:2px; border:2px
outset gray}

        </STYLE>
      </HEAD>
      <SCRIPT>
        <xsl:comment>
          <![CDATA[
function sort(field)
{
  sortField.value = field;
  <!-- set cursor to watch here? -->
  listing.innerHTML =
source.documentElement.transformNode(stylesheet);
}

```

```

]]>
    </xsl:comment>
</SCRIPT>
<SCRIPT for="window" event="onload">
    <xsl:comment>
        <![CDATA[
stylesheet = document.XSLDocument;
source = document.XMLDocument;
sortField = document.XSLDocument.selectSingleNode("//@order-by");
]]>
    </xsl:comment>
</SCRIPT>
<BODY>
    <TABLE width="100%" cellspacing="0">
        <TR>
            <TD class="bg">
                <H1>
                    ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ
                </H1>
            </TD>
        </TR>
        <TR>
            <TD class="bg" valign="top">
                <DIV id="listing">
                    <xsl:apply-templates match="ExtraApp"/>
                </DIV>
            </TD>
        </TR>
    </TABLE>
</BODY>
</HTML>
</xsl:template>
<xsl:template match="ExtraApp">
    <TABLE STYLE="background-color:white">
        <THEAD>
            <TD width="150">
                <DIV class="header">Επωνυμία</DIV>

```

```

</TD>
<TD width="100">
    <DIV class="header">Κωδικός Πελάτη</DIV>
</TD>
<TD width="120">
    <DIV class="header">Τρέχουσες πωλήσεις έτους</DIV>
</TD>
<TD width="130">
    <DIV class="header">Υπόλοιπο</DIV>
</TD>
<TD width="100">
    <DIV class="header">Τελευταία Είσπραξη</DIV>
</TD>
<TD width="100">
    <DIV class="header">Πιστωτικό όριο</DIV>
</TD>
<TD width="120">
    <DIV class="header">Πιστωτικά</DIV>
</TD>
</THEAD>
<xsl:for-each select="Extra">
    <TR>
        <TD>
            <DIV class="row">
                <xsl:value-of select="EPONYMIA"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="KODIKOS"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:apply-templates select="POLHSEIS"/>
            </DIV>
        </TD>
    </TR>
</xsl:for-each>

```

```

        <TD>
            <DIV class="row">
                <xsl:apply-templates select="ΥΠΟΛΟΙΠΟ"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="LAST"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:apply-templates select="ORIO"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:apply-templates select="PISTOTIKA"/>
            </DIV>
        </TD>
    </TR>
</xsl:for-each>
</TABLE>
</xsl:template>
<xsl:template match="POLHSEIS|ΥΠΟΛΟΙΠΟ|ORIO|PISTOTIKA">
    <xsl:eval>formatNumber(this.nodeTypeValue ,
"#,###,###")</xsl:eval>
</xsl:template>
</xsl:stylesheet>

```

### **FstockStyle.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
    <TABLE BORDER="1">
        <THEAD>
            <TH>Τύπος</TH>

```

```

        <TH>Μήκος</TH>
        <TH>Διαίρεση</TH>
    </THEAD>
    <xsl:for-each select="FINDSTOCK/seekstock" order-
by="EmvQuantity">
        <TR>
            <TD STYLE="padding-left:2em">
                <SPAN>
                    <xsl:value-of select="EmvName"/>
                </SPAN>
            </TD>
            <TD STYLE="padding-left:2em">
                <SPAN>
                    <xsl:value-of select="EmvQuantity"/>
                </SPAN>
            </TD>
            <TD STYLE="padding-left:2em text-align:center">
                <SPAN>
                    <xsl:value-of select="DIAIR"/>
                </SPAN>
            </TD>
        </TR>
    </xsl:for-each>
</TABLE>
</DIV>

```

### **OIKONOMIKA.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/W3C-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
    <xsl:template match="/">
        <HTML>
            <HEAD>
                <STYLE>
                    BODY {margin:0}
                    .bg {font:8pt Verdana; background-color:purple; color:white}
                    H1 {font:bold 14pt Verdana; width:100%; margin-top:1em}

```

```
.row {font:8pt Verdana; border-bottom:1px solid #CC88CC}
.header {font:bold 9pt Verdana; cursor:hand; padding:2px; border:2px
outset gray}
.up {background-color:#DDFFDD;}
.down {background-color:#FFDDDD;}
</STYLE>
</HEAD>
<SCRIPT>
  <xsl:comment>
    <![CDATA[
function sort(field)
{
  sortField.value = field;
  <!-- set cursor to watch here? -->
  listing.innerHTML =
source.documentElement.transformNode(stylesheet);
}
]]>
  </xsl:comment>
</SCRIPT>
<SCRIPT for="window" event="onload">
  <xsl:comment>
    <![CDATA[
stylesheet = document.XSLDocument;
source = document.XMLDocument;
sortField = document.XSLDocument.selectSingleNode("//@order-by");
]]>
  </xsl:comment>
</SCRIPT>
<BODY>
  <TABLE width="100%" cellspacing="0">
    <TR>
      <TD class="bg"/>
      <TD class="bg">
        <H1>
          ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ
        </H1>
```



```

        </TD>
    </TR>
    <TR>
        <TD class="bg" width="120" valign="top">
            <P>Πατήστε στις επικεφαλίδες των στηλών για
            αύξουσα ταξινόμηση</P>
            <P>Οι πελάτες που παρουσιάζουν αύξηση πωλήσεων
            χρωματίζονται με πράσινο και οι πτωτικοί με λευκό</P>
        </TD>
        <TD class="bg" valign="top">
            <DIV id="listing">
                <xsl:apply-templates match="EconomicData"/>
            </DIV>
        </TD>
    </TR>
</TABLE>
</BODY>
</HTML>
</xsl:template>
<xsl:template match="EconomicData">
    <TABLE STYLE="background-color:white">
        <THEAD>
            <TD width="80">
                <DIV class="header"
                onClick="sort('KLEIDI')">ΚΩΔΙΚΟΣ</DIV>
            </TD>
            <TD width="200">
                <DIV class="header"
                onClick="sort('EPONYMIA')">ΕΠΩΝΥΜΙΑ</DIV>
            </TD>
            <TD width="80">
                <DIV class="header"
                onClick="sort('change')">ΜΕΤΑΒΟΛΗ</DIV>
            </TD>
            <TD width="80">
                <DIV class="header"
                onClick="sort('TZIROS97')">ΤΖΙΡΟΣ1997</DIV>

```

```

</TD>
<TD width="80">
  <DIV class="header"
onClick="sort('TZIROS98')">ΤΖΙΡΟΣ1998</DIV>
</TD>
<TD width="80">
  <DIV class="header"
onClick="sort('POLHSEIS')">ΠΩΛΗΣΕΙΣ2000</DIV>
</TD>
<TD width="80">
  <DIV class="header" onClick="sort('Perc')">ΠΟΣΟΣΤΙΑΙΑ
ΜΕΤΑΒΟΛΗ 97 -98</DIV>
</TD>
</THEAD>
<xsl:for-each select="Economy" order-by="POLHSEIS">
  <TR>
    <xsl:for-each select="change">
      <xsl:if expr="this.nodeType > 0">
        <xsl:attribute name="class">up</xsl:attribute>
      </xsl:if>
    </xsl:for-each>
    <TD>
      <DIV class="row">
        <xsl:value-of select="KLEIDI"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row">
        <xsl:value-of select="EPONYMIA"/>
      </DIV>
    </TD>
    <TD>
      <DIV class="row" STYLE="text-align:right">
        <xsl:apply-templates select="change"/>
      </DIV>
    </TD>
  </TR>
</xsl:for-each>
</TD>

```

```

        <DIV class="row" STYLE="text-align:right">
            <xsl:apply-templates select="TZIROS97"/>
        </DIV>
    </TD>
    <TD>
        <DIV class="row" STYLE="text-align:right">
            <xsl:apply-templates select="TZIROS98"/>
        </DIV>
    </TD>
    <TD>
        <DIV class="row" STYLE="text-align:right">
            <xsl:apply-templates select="POLHSEIS"/>
        </DIV>
    </TD>
    <TD>
        <DIV class="row" STYLE="text-align:right">
            <xsl:apply-templates select="Perc"/>
        </DIV>
    </TD>
</TR>
</xsl:for-each>
</TABLE>
</xsl:template>
<xsl:template match="change|TZIROS97|TZIROS98|POLHSEIS">
    <xsl:eval>formatNumber(this.nodeTypeValue , "#,###,###")</xsl:eval>
</xsl:template>
<xsl:template match="Perc">
    <xsl:eval>formatNumber(this.nodeTypeValue , "#,###,###")</xsl:eval>%
    </xsl:template>
</xsl:stylesheet>

```

### **OrderCust.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
xmlns="http://www.w3.org/TR/REC-html40" result-ns="">
    <xsl:template match="/">

```

```

<HTML>
  <HEAD>
    <STYLE>
      BODY {margin:0}
      .bg {font:8pt Verdana; background-color:blue; color:white}
      H1 {font:bold 14pt Verdana; width:100%; margin-top:1em}
      .row {font:9pt Arial; border-bottom:1px solid #CC88CC}
      .header {font:bold 9pt Verdana; cursor:hand; padding:2px; border:2px
outset gray}

    </STYLE>
  </HEAD>
  <BODY>
    <TABLE width="100%" cellspacing="0">
      <TR>
        <TD class="bg">
          <H1>
            ΟΙΚΟΝΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΩΝ
          </H1>
        </TD>
      </TR>
      <TR>
        <TD class="bg" valign="top">
          <DIV id="listing">
            <xsl:apply-templates match="OrdersC"/>
          </DIV>
        </TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
</xsl:template>
<xsl:template match="OrdersC">
  <TABLE STYLE="background-color:white">
    <THEAD>
      <TD width="40">
        <DIV class="header">No Παραγγελίας</DIV>

```

```

</TD>
<TD width="50">
    <DIV class="header">Κωδικός Πελάτη</DIV>
</TD>
<TD width="100">
    <DIV class="header">Επωνυμία</DIV>
</TD>
<TD width="100">
    <DIV/>
</TD>
<TD width="80">
    <DIV class="header">Ημερ/νία Παραγγελίας</DIV>
</TD>
<TD width="80">
    <DIV class="header">Επιθυμητή Παράδοση</DIV>
</TD>
</THEAD>
<xsl:for-each select="OrdersCust" order-by="LastName">
    <TR>
        <TD>
            <DIV class="row">
                <xsl:value-of select="OrderID"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="CustomerID"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="LastName"/>
            </DIV>
        </TD>
        <TD>
            <DIV class="row">
                <xsl:value-of select="FirstName"/>
            </DIV>
        </TD>
    </TR>
</xsl:for-each>

```

```

        </DIV>
    </TD>
    <TD>
        <DIV class="row">
            <xsl:value-of select="OrderDate"/>
        </DIV>
    </TD>
    <TD>
        <DIV class="row">
            <xsl:value-of select="RequiredDate"/>
        </DIV>
    </TD>
</TR>
</xsl:for-each>
</TABLE>
</xsl:template>
<xsl:template
match="OrderID|CustomerID|LastName|FirstName|OrderDate|RequiredDate
"/>
</xsl:stylesheet>

```

### **ORDERSMHNA.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
    <H1>ΛΗΨΕΙΣ ΠΑΡΑΓΓΕΛΙΩΝ ΑΝΑ ΜΗΝΑ </H1>
    <TABLE BORDER="1">
        <THEAD>
            <TH>ΜΗΝΑΣ</TH>
            <TH>ΛΗΨΗ ΠΑΡΑΓΓΕΛΙΩΝ ΜΗΝΑ</TH>
        </THEAD>
        <xsl:for-each select="Ordersmhna/SALES" order-by="+ ΜΗΝΑΣ">
            <TR>
                <TD ALIGN="CENTER">
                    <SPAN>
                        <xsl:value-of select="ΜΗΝΑΣ"/>
                    </SPAN>
                </TD>

```

```

        <TD ALIGN="CENTER">
            <SPAN>
                <xsl:value-of select="ΠΛΗΘΟΣΜΗΝΑ"/>
            </SPAN>
        </TD>
    </TR>
</xsl:for-each>
</TABLE>
</DIV>

```

**SassiSales.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/WD-xsl">
    <xsl:for-each select="SalesSassi/bedates">
        Πωλήσεις σασσί από : <B>
            <xsl:value-of select="fidates"/>
        </B>
        έως : <B>
            <xsl:value-of select="ladates"/>
        </B>
    </xsl:for-each>
    <br/>
    <TABLE BORDER="1">
        <THEAD>
            <TH>Κωδικός</TH>
            <TH>Τύπος</TH>
            <TH>Πωλήσεις</TH>
        </THEAD>
        <xsl:for-each select="SalesSassi/SassiSales" order-by="+
SumOrders">
            <TR>
                <TD STYLE="padding-left: 1em">
                    <SPAN>
                        <xsl:value-of select="SassiID"/>
                    </SPAN>
                </TD>
                <TD STYLE="padding-left: 1em">

```

```

        <SPAN>
            <xsl:value-of select="SassiName"/>
        </SPAN>
    </TD>
    <TD STYLE="text-align:center">
        <SPAN>
            <xsl:value-of select="SumOrders"/>
        </SPAN>
    </TD>
</TR>
</xsl:for-each>
</TABLE>
</DIV>

```

### **TankSaStyle.xsl**

```

<?xml version="1.0" encoding="ISO-8859-7"?>
<DIV xmlns:xsl="http://www.w3.org/TR/W3-XSL1">
    <B> Πωλήσεις Δοχείων </B>
    <HR/>
    <TABLE BORDER="1">
        <THEAD>
            <TH>Τύπος Δοχείου </TH>
            <TH>Πωλήσεις</TH>
        </THEAD>
        <xsl:for-each select="TankData/Doxeia" order-by="DoxeioName">
            <TR>
                <TD STYLE="padding-left:2em">
                    <SPAN>
                        <xsl:value-of select="DoxeioName"/>
                    </SPAN>
                </TD>
                <TD STYLE="text-align:center">
                    <SPAN>
                        <xsl:value-of select="SumOfOrderID"/>
                    </SPAN>
                </TD>
            </TR>

```



```

        </xsl:for-each>
    </TABLE>
</DIV>

```

### **Ware.xsl**

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <HEAD>
        <TITLE>ΕΛΕΓΧΟΣ ΑΠΟΘΕΜΑΤΩΝ</TITLE>
        <STYLE TYPE="text/css">
          BODY {font-family:Tahoma,Arial,sans-serif;
            font-size:10pt; font-weight:normal;
            line-height:140%}
          .heading {font-family:Tahoma,Arial,sans-serif;
            font-size:16pt; font-weight:bold}
          .over {font-family:Tahoma,Arial,sans-serif;
            font-size:11pt; font-weight:bold}
          .under {font-family:Tahoma,Arial,sans-serif;
            font-size:11pt; color:red;
            font-weight:bold}
        </STYLE>
      </HEAD>
      <BODY>
        <DIV CLASS="heading">
          ΕΛΕΓΧΟΣ ΑΠΟΘΕΜΑΤΩΝ </DIV>
        <HR/>

        <xsl:for-each select="//MATSTOCK" order-by="-CODE">
          <xsl:apply-templates select="ΚΩΔΙΚΟΣ"/>
          <xsl:apply-templates select="ΚΑΤΗΓΟΡΙΑ"/>
          <xsl:apply-templates select="ΤΕΛΕΥΤΑΙΑ_ΠΑΡΑΓΓΕΛΙΑ"/>
          <xsl:apply-templates select="ΥΛΙΚΟ"/>
          <xsl:apply-templates select="ΑΠΟΘΕΜΑ"/>
          <xsl:apply-templates select="ΕΠΙΘΥΜΗΤΟ_ΑΠΟΘΕΜΑ"/>
          <xsl:apply-templates select="DIAFORA"/>
        <HR/>

```

```

        </xsl:for-each>
    </BODY>
</HTML>
</xsl:template>
<xsl:template match="">
    <xsl:node-name/>: <B>
        <xsl:value-of/>
    </B>
    <BR/>
</xsl:template>

```

```

<xsl:template match="DIAFORA">
    Η διαφορά μεταξύ αποθέματος και επιθυμητού αποθέματος είναι :<B>
        <xsl:eval>formatNumber(this.text, "#,###,##0")</xsl:eval>
    </B>μονάδες.<BR/>

```

```

<xsl:choose>
    <xsl:when match="*[$ $ 0]">
        <SPAN CLASS="under">
            <IMG SRC="under.gif" ALIGN="BOTTOM" HSPACE="5"/>
    ΠΡΟΣΟΧΗ : ΤΑ ΑΠΟΘΕΜΑΤΑ ΕΙΝΑΙ ΛΙΓΟΤΕΡΑ ΑΠΟ ΤΑ ΕΠΙΠΕΔΑ
    ΑΣΦΑΛΕΙΑΣ
        </SPAN>
        <BR/>
    </xsl:when>
    <xsl:otherwise>
        Τα αποθέματα είναι άνω του ορίου ασφαλείας<BR/>
    </xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

## *Αρχεία XML*

### **Employee.xml**

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<EMPLOYEES>
  <ΕΡΓΑΖΟΜΕΝΟΣ>
    <CODE>1</CODE>
    <TITLE>ΛΙΑΡΕΤΟΣ ΓΙΩΡΓΟΣ</TITLE>
    <CATEGORY>ΜΗΧΑΝΙΚΟΣ ΠΑΡΑΓΩΓΗΣ</CATEGORY>
    <PRDATE>16/02/96</PRDATE>
    <WAGE>220.000</WAGE>
    <ΤΜΗΜΑ>Παραγωγής</ΤΜΗΜΑ>
  </ΕΡΓΑΖΟΜΕΝΟΣ>
<ΕΡΓΑΖΟΜΕΝΟΣ>
  <CODE>2</CODE>
  <TITLE>ΡΕΤΟΠΟΥΛΟΣ ΓΙΑΝΝΗΣ</TITLE>
  <CATEGORY>ΜΗΧΑΝΟΛΟΓΟΣ ΜΗΧΑΝΚΟΣ</CATEGORY>
  <PRDATE>16/04/98</PRDATE>
  <WAGE>320.000</WAGE>
  <ΤΜΗΜΑ>ΤΕΧΝΙΚΟ</ΤΜΗΜΑ>
</ΕΡΓΑΖΟΜΕΝΟΣ>
<ΕΡΓΑΖΟΜΕΝΟΣ>
  <CODE>3</CODE>
  <TITLE>ΠΑΠΑΔΟΠΟΥΛΟΣ ΝΙΚΟΣ</TITLE>
  <CATEGORY>ΤΟΡΝΑΔΟΡΟΣ</CATEGORY>
  <PRDATE>12/09/90</PRDATE>
  <WAGE>340.000</WAGE>
  <ΤΜΗΜΑ>Παραγωγής</ΤΜΗΜΑ>
</ΕΡΓΑΖΟΜΕΝΟΣ>
</EMPLOYEES>
```

## ΑΝΑΦΟΡΕΣ

### ΒΙΒΛΙΟΓΡΑΦΙΑ

1. *Homer A.*, XML IE 5, εκδόσεις WROX, 1999
2. *Mc Grath S.*, XML by example, εκδόσεις Prentice Hall, 1998
3. *Spencer P.*, Professional XML design and implementation, εκδόσεις WROX, 1999
4. *Direnzo O., Duckett J. Graf J., Houle P., Jenkins T., Jones P., Kingsley-Hughes A., Kingsley-Hughes K., McQueen C., Mohr S.*, XML applications, εκδόσεις WROX, 1999

### Επιλεγμένα Web sites

1. *Asp 101*  
<http://www.asp101.com/home/home.asp>  
News, δείγματα, άρθρα, links.
2. *Asp alliance*  
<http://www.aspalliance.com/>  
Πολλά και χρήσιμα links σχετικά με την asp.
3. *Asp Developres network*  
<http://www.aspdeveloper.net/>  
Παραδείγματα και μαθήματα για asp
4. *Asp Town*  
<http://www.asptown.com/default.asp>  
Πηγές Asp
5. *Asp sites Resource guide*  
<http://www.aspsites.com/>  
Πλήρες site, πολλά άρθρα και οτιδήποτε μπορεί να προσφέρει ένα site του είδους του.
6. *Asp Zone.*  
<http://www.aspzone.com/>

Για προχωρημένους developers.

*7. Extensible Markup Language (XML)*

<http://www.w3.org/XML/>

Το site του W3C για την XML. Τυποποίηση , εξελίξεις, links, όλα τα πρότυπα των επεκτάσεων XML. Το εγκυρότερο όλων των site.

*8. IBM developer works*

<http://www.ibm.com/developer/xml/>

Το site της IBM για την XML. News, case studies, tools, άρθρα. Πλήρες web site.

*9. Learn Asp*

<http://www.learnasp.com/>

Μαθήματα και message boards

*10. MSDN Online: XML Developer Center*

<http://www.msdn.microsoft.com/xml/default.asp>

Ο δικτυακός τόπος της Microsoft για την XML

*11. Oasis*

<http://www.oasis-open.org/>

Η σελίδα του “Organization for the Advancement of Structured Information Standards” που συμπεριλαμβάνει και το πρότυπο XML.

*12. VBXML.COM*

<http://www.vbxml.com/default.asp>

XML community για Windows XML developers

*13. Wrox Web Developer*

<http://webdev.wrox.co.uk/>

Χρήσιμο site με βιβλιογραφία από τις εκδόσεις Wrox και δείγματα κώδικα καθώς και case studies

*14. Xml webring*

<http://www.vtopia.com/webring/>

Χρήσιμο xml webring

15. *Xml.com –Style matters*

<http://www.xml.com/>

Άρθρα, μαθήματα, FAQs, forums.

16. *XML: Extensible Markup Language*

<http://stars.com/Authoring/Languages/XML/>

Links, λογισμικό, άρθρα κ.λ.π από Web Developer's Virtual Library

17. *XML Information site*

<http://www.xmlinfo.com/>

Πληθώρα επιλογών, μπορεί να χαρακτηριστεί πολύ χρήσιμο.

18. *Xml Organization*

<http://www.xml.org/>

Εκπαίδευση, πληροφορίες για την XML, news

19. *Xml – software*

<http://www.xmlsoftware.com/>

Απαραίτητο site για τον προγραμματιστή XML καθώς παρέχει πληθώρα εργαλείων.

20. *XML Zone*

<http://www.xml-zone.com/>

News, άρθρα, links.

## **Άρθρα**

1. *Awai Michael*, Applying XML and ASP to Create a Web Application Framework

<http://www.asptoday.com/articles/20000105.htm>

2. *Bosak Jon, Bray Tim*, XML and the Second-Generation Web

<http://www.sciam.com/1999/0599issue/0599bosak.html>

3. *Cagle Kurt*, Transform Your Data With XSL

<http://www.xmlmag.com/upload/free/features/xml/1999/01win99/kc2win99/kc2win99.asp>

4. *Cagle Kurt*, Create Extensible Web Pages With XML and CSS

<http://www.xmlmag.com/upload/free/features/xml/1999/01win99/kc3win99/kc3win99.asp>

5. *Cagle Kurt*, Maintain Large Databases With XML Servers

[http://www.inquiry.com/techtips/xml\\_pro/10min/10min0200/10min0200.asp](http://www.inquiry.com/techtips/xml_pro/10min/10min0200/10min0200.asp)

6. *Cagle Kurt*, Implement Sophisticated Tables in IE5

<http://www.devx.com/upload/free/features/webbuilder/1999/wb0599/kc0599/kc0599.asp>

7. *Cooke Kevin*, 30/09/1998 Introduction to Active Server Pages

<http://hotwired.lycos.com/webmonkey/98/39/index2a.html>

8. *Heinemann Charles*, Internet Explorer 5 and XML

<http://msdn.microsoft.com/xml/articles/xml110498.asp>

9. *Heinemann Charlie*, Describe Your Data 20/07/1999

<http://msdn.microsoft.com/xml/articles/xml072099.asp>

10. *Heinemann Charlie*, Schemas: Tapping the Potential Within 16/08/999

<http://msdn.microsoft.com/xml/articles/xml081699.asp>

11. *Heinemann Charlie*, Using XSL to Sort and Filter Your Data 03/ 999

<http://msdn.microsoft.com/xml/articles/xml020899.asp>

12. *Heinemann Charles*, Happy Days Are Here Again: Posting XML to the Server 22/04/1999

<http://msdn.microsoft.com/xml/articles/xml071798.asp>

13. *Isaacs Scott*, Building Documents with XML, XSL, and CSS

<http://www.siteexperts.com/tips/xml/ts01/page1.asp>

14. *Isaacs Scott*, XML-Based Survey Server

<http://www.siteexperts.com/tips/xml/ts04/page1.asp>

15. *Isaacs Scott*, Building Smart Pages with ASP, XML and XSL

<http://www.siteexperts.com/tips/xml/ts02/page1.asp>

16. *Jones Russell*, Take Advantage of XML Using VB and ASP

<http://www.devx.com/upload/free/features/vbpj/1999/12dec99/id1299/id1299.asp>

17. *McBee Alan*, ASP Technology and the XML DOM 20/09/1999

<http://msdn.microsoft.com/xml/articles/xml092099.asp>

18. *Randell Brian*, A Beginner's Guide to the XML DOM , 10/1999

<http://msdn.microsoft.com/xml/articles/beginner.asp>

19. *Rich Jonathan*, XML and the IT architect

<http://www.sunworld.com/swol-06-1999/swol-06-itarchitect.html>

20. *Simon St. Laurent*, Describing your data : DTDs and XML schemas

<http://xml.about.com/compute/xml/gi/dynamic/offsite.htm?site=http://xml.com/pub/1999/12/dtd/index.html%3Fwwwrrr%5F19991201.txt>

21. *Tidwell Doug*, XML and how it will change the Web 11/ 1999

<http://www-4.ibm.com/software/developer/library/xml-web/index.html>

22. *Walsh Norman*, Schemas for XML 01/07/1999

[http://xml.about.com/compute/xml/gi/dynamic/offsite.htm?site=http://xml.com/xml/pub/1999/07/schemas/index.html%3Fwwwrrr\\_990701.txt](http://xml.about.com/compute/xml/gi/dynamic/offsite.htm?site=http://xml.com/xml/pub/1999/07/schemas/index.html%3Fwwwrrr_990701.txt)

### **Newsgroups :**

1. windows.inetexplorer.ie5.programming.xml

<http://msdnnews.microsoft.com/default.asp>

2. alt.activeserverpages



### 3. Intranet eXchange

<http://idm.internet.com/ix/index.html>

## Ηλεκτρονικές Λίστες

1. <http://www.vbxml.com/>

[vbxml@onelist.com](mailto:vbxml@onelist.com)

2. <http://www.asplists.com/asplists/aspxml.asp>

[aspxml@ls.asplists.com](mailto:aspxml@ls.asplists.com)

3. <http://www.projectcool.com/>

[xml@mercury.projectcool.com](mailto:xml@mercury.projectcool.com)

4. <http://www.asplists.com/asplists/aspdatabases.asp>

[aspdatabases@ls.asplists.com](mailto:aspdatabases@ls.asplists.com)

## Εφημερίδες /Περιοδικά

1. *Intranet journal*

<http://www.intranetjournal.com/>

2. *XML Magazine*

<http://www.xmlmag.com/>