ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΑΝΟΙΚΤΑ ΑΚΑΔΗΜΑΙΚΑ ΜΑΘΗΜΑΤΑ

# Ειδικά Θέματα Βάσεων Δεδομένων

## Ενότητα 14: Εισαγωγή στην XML

Δρ. Τσιμπίρης Αλκιβιάδης

Τμήμα Μηχανικών Πληροφορικής ΤΕ

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

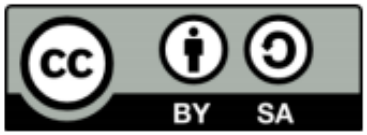
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.

- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

# Ενότητα 14

## XML

Δρ. Τσιμπίρης Αλκιβιάδης

# Περιεχόμενα ενότητας

- Εισαγωγή στην XML
- Τι είναι XML
- Ιστορική Αναδρομή
- Πλεονεκτήματα
- Καινοτομίες
- Δομή της XML
- Στοιχεία XML
- Χρήση της XML
- Χαρακτηριστικά
- DTD Παράδειγμα
- Παρουσίαση XML Εγγραφών
- Λειτουργίες XML
- XML QUERY Γλώσσες
- XML Native ΒΔ Βασισμένη σε πίνακα χαρτογράφησης
- Αντικειμενοστραφής χαρτογράφηση
- Ερωτήσεις XML Σε Σχεσιακές ΒΔ
- Παραδείγματα Ευρετηρίου
- Σύστημα Αρίθμησης

# Σκοποί ενότητας

Ο σκοπός της ενότητας αυτής είναι καταρχήν η εισαγωγή στην XML και μέσα από μια ιστορική αναδρομή να παρουσιαστεί η ανάγκη που οδήγησε στον ορισμό της. Παρουσιάζονται τα πλεονεκτήματα και οι καινοτομίες που παρέχει. Ο ορισμός της δομή της XML με τα απαραίτητα στοιχεία σχήματα (DTD) XML μπορούν να την κάνουν εργαλείο χρήσιμο για τη μεταφορά δεδομένων μεταξύ ομοίων ή και διαφορετικών RDBMS.

# Εισαγωγή στην XML

- Με την τεράστια εξάπλωση του παγκόσμιου ιστού, η ανάγκη για ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων και πλατφορμών έγινε επιτακτική
- Το κύριο πρόβλημα είναι ότι η μορφή και ο τύπος των δεδομένων ποικίλλει. Μπορεί να είναι αρχεία κειμένου, δεδομένα βάσεων δεδομένων, μεταδεδομένα(metadata) κτλ.
- Επιτακτική ανάγκη για ένα κοινό πρότυπο αναπαράστασης και ανταλλαγής δεδομένων, κοινό για όλες τις πλατφόρμες

# Τι είναι XML

- Το XML αποτελεί συντομογραφία του **eXtensive Markup Language**
- Σύνολο απο κανόνες για την δημιουργία ετικετών (tags) που **περιγράφουν** τα δεδομένα ενός εγγράφου καθώς και προσδιορίζουν και τα διάφορα μέρη απο τα οποία αποτελείται ένα έγγραφο
- Οι ετικέτες δεν είναι προκαθορισμένες, τις ορίζουμε εμείς
- Μεταγλώσσα με την οποία μπορούμε να ορίσουμε άλλες γλώσσες σήμανσης

# Ιστορική Αναδρομή

- Η ανάπτυξη του **XML** ξεκίνησε το 1996
- Εντάχθηκε στο **W3C** το 1998
- Βασίζεται στην **SGML (Standard Generalized Markup Language**) και είναι ένα *υποσύνολό* της. Διατήρησε τα λειτουργικά χαρακτηριστικά της, αλλά απέβαλε τα στοιχεία εκείνα που την έκαναν **δύσχρηστη** στο προγραμματισμό
- Ευρέος γνωστή στον τομέα του διαδικτύου

# Πλεονεκτήματα

- Εύκολο στην κατανόηση
- Κοινό πρότυπο μεταξύ διαφορετικών πλατφορμών
- Αυτο-περιγραφική γλώσσα
- Αποθήκευση σε ASCII κείμενο
- Ευελιξία στη δομή καθώς ο καθένας δημιουργεί όσες και όποιες ετικέτες θέλει
- Ευανάγνωστα αρχεία
- Οι περισσότερες εφαρμογές υποστηρίζουν την εξαγωγή και εισαγωγή στοιχείων από έγγραφα XML
- Υποστήριξη απο πληθώρα τεχνολογιών και εργαλείων

# Καινοτομίες

Σημαντικότερες *αλλαγές* που έφερε το XML:

- Δεδομένα
- Αρχιτεκτονική
- Λογισμικό

# Καινοτομίες

**Δεδομένα**

- Αποδέσμευση των δεδομένων από τις εφαρμογές
- Δημιουργία επιχειρηματικών λεξικών
- Ανάπτυξη της B2B επικοινωνίας

**Αρχιτεκτονική**

- Ανάπτυξη κατανεμημένων υπολογιστικών συστημάτων
- Χρήστη της XML και του διαδικτύου (TCP/IP) για επικοινωνία και ανταλλαγή δεδομένων και πληροφοριών

# Καινοτομίες

**Λογισμικό**

- Με τη χρήση της XML επεκτείνεται η δυνατότητα επικοινωνίας διαφόρων υποπρογραμμάτων και εφαρμογών μεταξύ τους
- Προγράμματα βασισμένα σε modules για τη λύση συγκεκριμένων προβλημάτων
- Απλότητα και ευελιξία

# Δομή του XML

- Ένα XML έγγραφο μοιάζει με ένα HTML έγγραφο
  - Το **XML** σχεδιάστηκε για να **περιγράφει** δεδομένα
  - Η **HTML** σχεδιάστηκε για να **εμφανίζει** δεδομένα
- Αποτελείται από tags τα οποία είναι υποχρεωτικό να κλείνονται (σε αντίθεση με την HTML)
  - π.χ., <author>George R. R. Martin</author>
- Επιτρέπονται **άπειρα** επίπεδα εμφωλευμένων tag
- **Απαγορεύεται** οι ετικέτες να ξεκινούν με 'XML' είτε σε πεζά είτε σε κεφαλαία

# Στοιχεία XML

- **Element:** Το βασικό δομικό στοιχείο
  - <book> Game of Thrones </book>
- **Attribute:** Ιδιότητα ενός Element
  - <book author=”George R. R. Martin”> Game of Thrones </book>
- Τα attributes μπορούν να γραφούν και με τη μορφή εμφωλευμένων Element
  - π.χ.,
  - <book>
- <title>Game of Thrones </title>
  - <author>George R. R. Martin</author>
- </book>

# Στοιχεία XML

- **Entity:** Οι οντότητες είναι αλφαριθμητικά που χρησιμοποιούνται ως συντομογραφίες άλλων αλφαριθμητικών

  **π.χ., <!Entity message “Hello World”>**

- Με αυτόν τον τρόπο, όταν γράφουμε **&message** ισοδυναμεί με “**Hello World**”

# Χρήση της XML

- **Επιχειρηματικά δεδομένα** (πωλήσεις, τιμολόγια, παραγγελίες)
- **Στοιχεία απο βάσεις δεδομένων**
  - Όλα τα σύγχρονα RDBMS υποστηρίζουν εξαγωγή και εισαγωγή απο XML
- **Οικονομικά δεδομένα**
- **Human Resources XML (HR-XML)**

# Χαρακτηριστικά

- Δεν κάνει υπόθεση για τον τρόπο παρουσίασης των εγγράφων
- Δεν ορίζονται τύποι στα δεδομένα της
- Δεν υπάρχει περιορισμός στον τρόπο μετάδοσης των εγγράφων
- Περιγράφει μόνο τη δομή και το περιεχόμενο των δεδομένων και όχι τον τρόπο παρουσίασης τους(Αντίθετα η HTML περιγράφει τον τρόπο παρουσίασης κάποιων δεδομένων)
- Οι ετικέτες του εγγράφου συνήθως σχετίζονται με την οντότητα στην οποία αναφέρονται

# DTD Παράδειγμα

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE movie[
<!ELEMENT movie(title, director)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
]>
<movie>
    <title>The Lord of the Rings: The Two Towers</title>
    <director>Peter Jackson</director>
</movie>
```

# Παρουσίαση XML Εγγραφών

Η XML δεν ορίζει τον τρόπο με τον οποίο θα παρουσιαστούν τα δεδομένα που περιγράφει

- Για την παρουσίαση, χρησιμοποιούνται άλλες γλώσσες ή εργαλεία
  - **CSS** (Cascading Style Sheets)
  - **XSL** (eXtensive Style sheets Language)
  - **XForms**
- Μειονεκτήματα CSS
  - Απαιτείται DTD
  - Εξαρτάται από τον browser

# Παρουσίαση XML Εγγραφών

- Πλεονεκτήματα XSL
  - Αλλαγή στη σειρά εμφάνισης των στοιχείων
  - Ανεξάρτητο browser
  - Περισσότερες λειτουργίες και δυνατότητες
- XForms
  - Αντικατάσταση των HTML φορμών
  - Διαχωρισμός εμφάνισης και λειτουργικότητας
  - Αποτέλεσμα σε XML

# Επεξεργασία

- Κάθε XML έγγραφο μπορεί να θεωρηθεί ως μια πηγή πληροφορίας δεδομένων
- Πρέπει να υποστηρίζονται πράξεις αναζήτησης πληροφορίας
- Κάθε XML έγγραφο μπορεί να θεωρηθεί ως ένα μη κατευθυνόμενο δέντρο

# Λειτουργίες XML

Οι λειτουργίες που θα πρέπει να υποστηρίζονται σε ένα XML έγγραφο είναι:

- Εύρεση σχέσης πατέρα - γιου
- Εύρεση σχέσης προγόνου - απόγονου
- Έλεγχος τιμής χαρακτηριστικού
- Καμία, μία ή περισσότερες εμφανίσεις ενός κόμβου
- Twig patter matching
- Ταίριασμα υποδέντρου

# XML QUERY Γλώσσες

Γλώσσες αναζήτησης

- XPath
- XQuery
- XPointer

XPath

- Δημιουργεί μια δενδρική δομή απο το XML έγγραφο
- Εκφράσεις
- Πατέρας-γιος:‘/’
- Πρόγονος - Απόγονος : ‘//’
- Attribute: ‘@’
- Book/author[@name=”John”]

Μία ή περισσότερες εμφανίσεις book/chapter+/title

# XML Native ΒΔ

Πολλές επειχειρήσεις χρησιμοποιούν την XML για αποθήκευση των επιχειρηαμτικών τους δεδομένων:

- Ημιδομημένα δεδομένα (semistructured data)
- Ανάπτυξη ολοκληρωμένων συστημάτων διαχείρησης
- XML εγγράφων (XML native databases)
- XML native βάσεις δεδομένων
- Συστήματα διαχείρισης XML εγγράφων, τα οποία
- Συστατικά:χρησιμοποιούν δικούς τους τρόπους αποθήκευσης, επεξεργασίας και απεικόνισης XML έγγράφων και δεδομένων
- Τρόπος αποθήκευσης
- Συλλογές XML εγγράφων
- Τρόπος εκτέλεσης ερωτημάτων

# Αποθήκευση XML εγγράφων

**Φυσικά αρχεία**

- Αργή πρόσβαση
- Απουσία δομών δεικτοδότησης

**Σχεσιακές βάσεις δεδομένων (RDBMS)**

- Ανάγκη μετατροπής σε πίνακες και στήλες
- Ύπαρξη δομών δεικτοδότησης
- Χρήση SQL για ερωτήσεις
- Ανάγκη DTD Schema

# XML Native ΒΔ

- Για να αποθηκευτεί ένα XML έγγραφο σε μια βάση δεδομένων, πρέπει το schema του να μετατραπεί σε ένα ανάλογο schema βάσης δεδομένων
- Διαχωρισμός elements, attributes και κειμένου.
- Δεν λαμβάνουμε υπόψιν την σειρά των elements και attributes
- Είδη χαρτογράφησης
  - Βασισμένη σε πίνακα
  - Αντικειμενοστραφής

# Βασισμένη σε πίνακα χαρτογράφησης

- Ένα XML έγγραφο μπορεί να αποθηκευτεί είτε ως ένας πίνακας είτε ως πολλοί
- Ανάλογα με το λογισμικό που χρησιμοποιείται, τα στοιχεία αποθηκεύονται
- Απαιτείται το XML έγγραφο να έχει την εξης μορφή:

```
<database>
        <table>
            <row>Onoma</row>
            <row>Hmerominia</row>
        </table>
</database>
```

# Βασισμένη σε πίνακα χαρτογράφησης

Τα attributes αποθηκεύονται αυτόματα ως στήλες στους αντίστοιχους πίνακες:

**Πλεονεκτήματα**

- Εύκολη υλοποίηση
- Δεν απαιτείται DTD

**Μειονεκτήματα**

- Δεν υποστηρίζονται όλα τα XML έγγραφα( πολλαπλές εμφωλεύσεις)

# Αντικειμενοστραφής χαρτογράφηση

- Χρησιμοποιείται σχεδόν σε όλες τις σύγχρονες σχεσιακές βάσεις δεδομένων
- Απαιτείται DTD σχήμα
- Οι τύποι στοιχείων και οι ιδιότητες τους δημιουργούνται ως κλάσεις
- Τα απλά στοιχεία (που έχουν μόνο τιμή κειμένου) διαμορφώνονται ως scalar ιδιότητες
- Οι κλάσεις αποθηκεύονται σε πίνακες και οι scalar ιδιότητες σε στήλες των αντίστοιχων πινάκων
- Οι ιδιότητες χαρτογραφούνται σε ζευγάρια πρωτεύοντος/ξένου κλειδιού
- Πλεονεκτήματα
  - Υποστηρίζονται όλα τα XML έγγραφα
  - Διατηρούνται οι ιδιότητες των πεδίων
  - Μπορούμε να ανακτήσουμε το DTD σχήμα απο το σχήμα δεδομένων

# Αντικειμενοστραφής χαρτογράφηση: παράδειγμα

<!ELEMENT Order (OrderNum, Date, CustNum, Item*)>
    <!ELEMENT OrderNum (#PCDATA)>
    <!ELEMENT CustNum (#PCDATA)>

<!ELEMENT Item (ItemNum, Quantity, Part)>
    <!ELEMENT ItemNum (#PCDATA)>
    <!ELEMENT Date (#PCDATA)>
    <!ELEMENT Quantity (#PCDATA)>

    <!ELEMENT Part (PartNum, Price)>
    <!ELEMENT PartNum (#PCDATA)>
    <!ELEMENT Price (#PCDATA)>

# Αντικειμενοστραφής χαρτογράφηση: παράδειγμα 2

**Βήμα 1**

Για κάθε σύνθετο στοιχείο, δημιουργείται ένας πίνακας και ένα αυτόματο πρωτεύον κλειδί πίνακα

Π.χ Δημιουργία Πινάκων Order, Item, Part και των αντίστοιχων PK: *OrderPK, ItemPK, PartPK*

**Βήμα 2**

Για κάθε απλό στοιχείο δημιουργείται μια στήλη στον αντίστοιχο πίνακα

Π.χ Δημιουργία των στηλών

OrderNum, Date, CustNum (πίνακας Order)

ItemNum,Quantity (πίνακας Item)

PartNum, Price (πίνακας Part)

# Αντικειμενοστραφής χαρτογράφηση: παράδειγμα 2

**Βήμα 3**

Παραγωγή ξένων κλειδιών για αναφορά στα σύνθετα στοιχεία

Π.χ Δημιουργία ξένου κλειδιού OrderFK (πίνακας Item), ItemFK(πίνακας Part)

# Ερωτήσεις XML Σε Σχεσιακές ΒΔ

- Για την επεξεργασία αποθηκευμένων XML εγγράφων υπάρχουν δύο τεχνικές
- Χρήση SQL και μετατροπή των αποτελεσμάτων σε XML
- Χρήση XML Query γλώσσας και μετατροπής της σε SQL statements
- Ευρετήρια:
  - Εγγενή ευρετήρια της βάσης
  - Πρόσθετα ευρετήρια βελτιστοποιημένα για XML

# Παράδειγμα Ευρετηρίου 1

Ένα απλό παράδειγμα ευρετηρίου είναι το 1 - index

Στο index αυτό όλοι οι κόμβοι ενός XML δέντρου ομαδοποιούνται ανάλογα με το path τους απο το root

Παράδειγμα

<βάλε εικόνα>

# Παράδειγμα Ευρετηρίου 2

- Στα φύλλα του ευρετηρίου αποθηκεύονται δείκτες οι οποίοι δείχνουν σε όλους τους κόμβους με κοινό path. Ο τύπος των δεικτών αυτών εξαρτάται με τον τρόπο αποθήκευσης του XML εγγράφου
- Το ευρετήριο αυτό είναι συνήθως πολύ μικρό σε μέγεθος και φορτώνεται στην μνήμη. Έτσι όταν έχουμε ένα query, κάνουμε αναζήτηση στο ευρετήριο, βρίσκουμε τους επιθυμητούς κόμβους (μέσω των δεικτών) και τους επιστρέφουμε

# Σύστημα Αρίθμησης

- Πολλά συστήματα επεξεργασίας XML εγγράφων, χρησιμοποιούν (σε συνδυασμό με ευρετήρια) κάποια συστήματα αρίθμησης των κόμβων ενός XML εγγράφου
- Γρήγορη εύρεση των δοκιμών σχέσεων μεταξύ δύο κόμβων
- Δύο κύριες κατηγορίες
  - Αριθμητικά συστήματα
  - Prefix-based συστήματα
- Αριθμητικά συστήματα
  - Βασίζονται κυρίως στις preorder και postorder labels του κάθε κόμβου(με διάφορες παραλλαγές)
- Prefix-based συστήματα
  - Βασίζονται στην ιδέα οτι η label ενός κόμβου είναι η label του πατέρα του + <κάτι άλλο>

# Σύστημα Αρίθμησης: Παράδειγμα

- Σε κάθε κόμβο αναθέτουμε ένα ζεύγος τιμών **<order, size>**
- Το order είναι ανάλογο της preorder σειρά του κόμβου, ενώ το size είναι τέτοιο ώστε ο κόμβος να περιέχει όλους τους απογόνους του
- Έστω δύο κόμβοι x, y. Τότε ο x είναι πρόγονος του y, αν και μόνο άν

  ***order(x) < order(y) + size(y) < order(x) + size(x)***

  **<Βάλε εικόνα>**