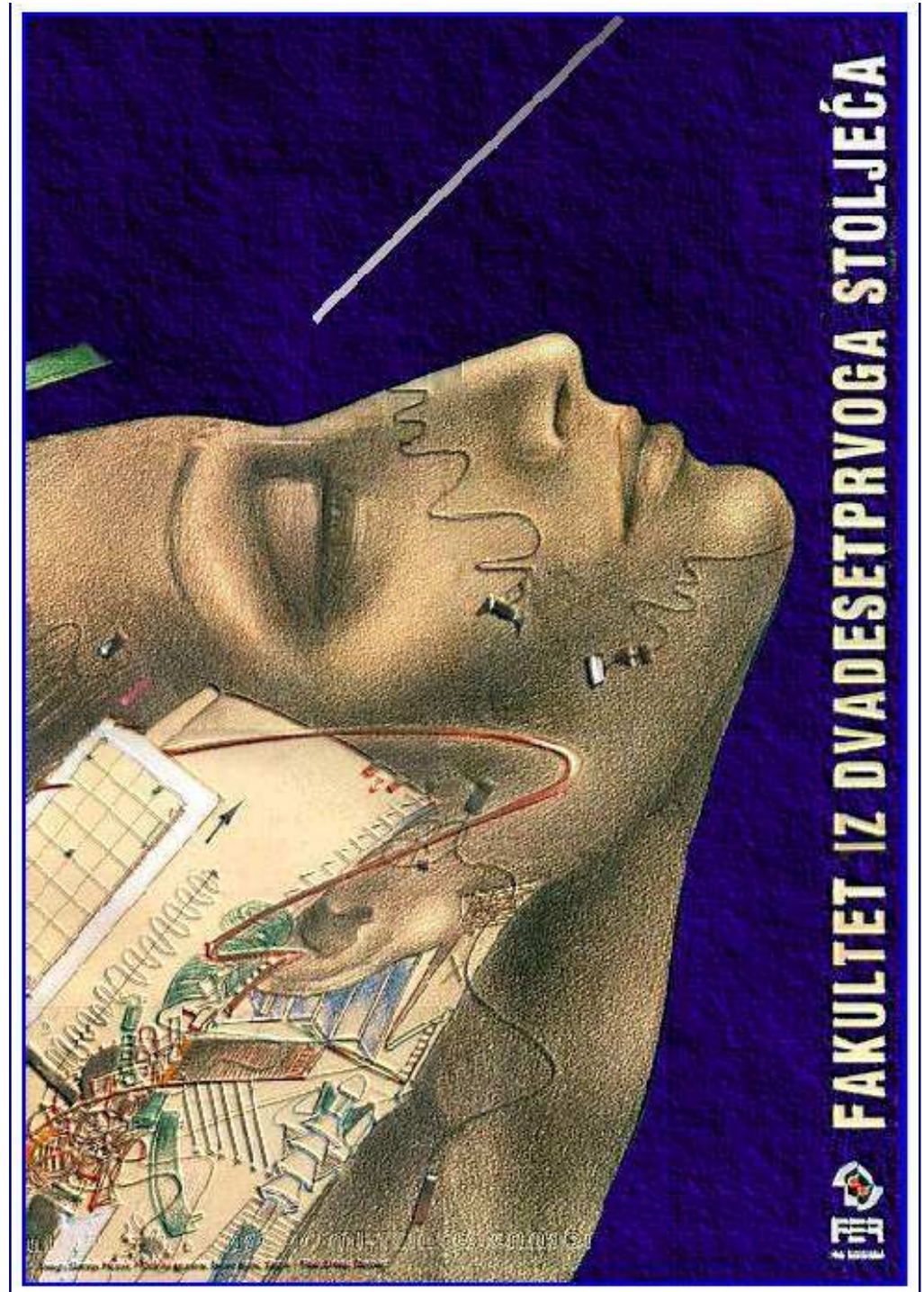


# Napredni modeli i baze podataka

Predavanja

## 6. XML baze baze podataka

Listopad 2008.



# Sadržaj

---

- XML – što je i čemu služi
- Pohrana XML dokumenata
- XML i RDB
- XML enabled BP
- Izvorne (Native) XML baze podataka
- XQuery

# XML

---

- EXtensible Markup Language
- Definirao ga je WWW Consortium (W3C)
- Koristi oznake(tags) kao i HTML
- Omogućuje opisivanje i pohranu polustrukturiranih dokumenata
- Služi za opisivanje (značenja?) podataka
- XML dokumenti imaju oznake (tags) koje dodatno opisuju dijelove dokumenta - samodokumentiranje
- Npr. <naslov> XML </naslov> <slide> Što je XML...</slide>
- Nema predefinirane oznake - svatko može definirati svoje vlastite oznake i opisati kako će se s tim oznakama rukovati pri ispisu

# Što je XML

---

- Strukturirani tekst – u obliku stabla
- Neovisan o računalnoj platformi – može se nalaziti bilo gdje
- Ne radi ništa
- Služi za pohranu i razmjenu informacija
- Podaci iz jednog XML dokumenta mogu se prikazati na različite načine
  - svi podaci ili samo neki, različiti oblik, poredak
  - prikaz na različitim uređajima –PC, ručna računala, mobitel, uređaji za osobe s posebnim potrebama, ...

# Čemu služi XML

---

- Odvajanje podataka od prezentacije – fleksibilnost
  - isti podaci mogu se prikazati na različite načine
- Razmjena podataka među heterogenim sustavima
  - mogućnosti dodavanja novih oznaka i kreiranja ugnježđenih struktura učinile su XML idealnom strukturom za razmjenu
- Pohrana podataka
- Pretraživanje podataka

# Primjer

---

```
<?xml version="1.0" encoding="utf-8"?>
<poruka>
  <odKoga>Hrvojka Marković</odKoga>
  <kome>Mirta Baranović</kome>
  <naslov>Uvid u domaću zadaću</naslov>
  <tekst>Stvarno ne znam u čemu je problem, ali meni
    program izbacuje dobre rezultate a ne navedene na
    ahyco-u. Možda sam stvarno nešto krivo prepisala ali
    kako više ne mogu ući u uvid koda, to ne mogu ni
    provjeriti. Bila bih Vam zahvalna kada biste provjerili i
    obavjestili me u čemu je problem.
  </tekst>
</poruka>
```

# Elementi i atributi

---

- Podaci se mogu pohraniti u elementima ili atributima.

## Primjer:

```
<osoba>
  <spol>ženski</spol>
  <ime>Jasna</ime>
  <prezime>Novak</prezime>
</osoba>
```

```
<osoba spol="ženski">
  <ime>Jasna</ime>
  <prezime>Novak</prezime>
</osoba>
```

Atributi se opisuju parom `ime="vrijednost"` unutar početne oznake elementa

Element može imati više atributa, ali se svaki atribut može pojaviti samo jednom:

```
<osoba spol="ženski" status="student">
```



# Valjan XML dokument (eng. valid)

---

- Valjan XML dokument je
  - dobro oblikovan (well formed)
  - udovoljava pravilima koja definira DTD (Document Type Definiton) ili XML Shema (XSD)
- Dobro oblikovan XML dokument
  - ima korijenski element
  - svaki element ima početnu i krajnju oznaku
  - u oznakama se razlikuju mala i velika slova
  - elementi moraju biti pravilno ugniježđeni (nije dozvoljeno preplitanje)
  - atributi moraju biti navedeni unutar navodnika
  - moguće je ugnježđivanje elemenata unutar elemenata



# Je li XML dokument baza podataka? (1)

---

- BAZA PODATAKA je skup podataka koji su pohranjeni i organizirani tako da mogu zadovoljiti zahtjeve korisnika.

*(M. Vetter, 1981.)*

- BAZA PODATAKA je skup međusobno povezanih podataka, pohranjenih zajedno, uz isključenje bespotrebne zalihosti (redundancije), koji mogu zadovoljiti različite primjene. Podaci su pohranjeni na način neovisan o programima koji ih koriste. Prilikom dodavanja novih podataka, mijenjanja i pretraživanja postojećih podataka primjenjuje se zajednički i kontrolirani pristup. Podaci su strukturirani tako da služe kao osnova za razvoj budućih primjena.

*(J. Martin, 1979.).*

# Je li XML dokument baza podataka? (2)

---

- XML dokument sadrži
  - podatke
  - metapodatke
- Metapodaci su sadržani u strukturi elemenata, atributima i hijerhiji elemenata
- XML elementi i atributi opisuju svojstva podataka – usporedivo s tablicama i atributima u relacijskoj bazi podataka
- XML dokument hijerarhijskom strukturom opisuje veze i odnose između različitih "kategorija podataka" – usporedivo s odnosima između tablica u relacijskoj bazi podataka

# Je li XML dokument baza podataka? (3)

---

- XML dokument je baza podataka u onolikoj mjeri u kojoj je to bilo koji drugi tekstualni dokument
  
- XML dokument NIJE baza podataka ali ima određene prednosti nad ostalim oblicima tekstualnih dokumenta
  - Sam sebe opisuje
  - Prenosiv je (unicode)
  - XML schemom (DTD ili XML Schema) može se precizno definirati struktura dokumenta
  - Postoje upitni jezici za rad s XML dokumentima (XPath, XQuery, XML\_QL, QUILT,...)
  - Postoje programska razvojna sučelja (SAX, DOM, JDOM)

# Pohrana XML dokumenata

---

- Dvije su osnovne mogućnosti pohrane XML dokumenata:
  - Datotečni sustav
  - Baza podataka
  
- Mogućnosti pohrane XML dokumenata u bazi podataka:
  - Relacijske baze podataka
  - XML baze podataka
    - XML- enabled baze podataka
    - Izvorne (native) XML baze podataka (NXD)

# XML dokumenti u datotečnom sustavu

---

- XML dokumenti se čuvaju kao datoteke u datotečnom sustavu
- Postoje alati za:
  - pretraživanje i prezentaciju u drugom formatu (DOM, XSLT, itd)
  - indeksiranje (specifične implementacije – nije uobičajeno svojstvo operacijskog sustava)
- Prihvatljivo za mali skup XML dokumenata u jednokorisničkom okruženju
- Nedostaci:
  - Indeksiranje
  - Sigurnost
  - Transakcije i integritet podataka
  - Višekorisnički rad

# XML u relacijskoj bazi podataka

---

- 2 načina preslikavanja XML → RDB:
  - jedan XML dokument u jednom polju
  - jedan XML dokument u više tablica

# XML u relacijskoj bazi podataka

## - jedan XML dokument u jednom polju

---

- cijeli XML dokument se sprema u jedno polje - BLOB/CLOB (Binary/Character Large Object)
  - ✗ DBMS “nije svjestan” da se radi o XML dokumentu pa ne nudi nikakve mehanizme za rad sa XML-om
  - ✗ sporo pretraživanje (potrebni posebni moduli jer XML dokumenti nisu indeksirani)
  - ✗ DBMS obično nema alat za postavljanje upita nad XML dokumentima - dohvat podataka nepraktičan i neefikasan
  - ✗ izmjena se obavlja nad cijelim dokumentom
  - ✓ spremanje u bazu je vrlo brzo i jednostavno
  - ✓ zadržava hijerarhijsku strukturu i originalni sadržaj XML-a



# XML u relacijskoj bazi podataka

## - jedan XML dokument u više tablica

---

- Ideja je iskoristiti prednosti relacijskog modela
- Potrebno definirati preslikavanje između XML scheme i relacijske sheme
- Struktura XML dokumenta se razlaže u tablice
- ✗ može biti vrlo složeno i sporo
- ✗ gubi se hijerarhija opisana XML-om
- ✓ dopušta pretraživanje prema kriterijima
- Dvije najznačajnije tehnike preslikavanja XML-RDB
  - relacijsko preslikavanje (table-based mapping)
  - objektno-relacijsko preslikavanje (object-relational mapping)

# Relacijsko preslikavanje (1)

---

- Struktura XML dokumenta mora biti "pravilnog" oblika
- XML dokument se razlaže na jednu ili više tablica
- Vrlo ograničene mogućnosti primjene zbog strogog formata kojeg XML dokument mora poštivati
- Koriste ga proizvodi koji služe za prijenos/razmjenu podataka između XML dokumenata i RDB
- Ovisno o proizvodu moguće je specificirati pohranjuje li se atribut iz RDB kao element ili kao atribut
- Obično se metapodaci RDB koriste za imenovanje elemenata/atributa u XML dokumentu
- Nije prikladno za nestrukturirane dokumente s “mješovitim” sadržajem (element sadrži i tekst i druge element i attribute)

# Relacijsko preslikavanje (2)

- XML sljedeće strukture može se jednostavno preslikati u relacijski model

```
<bazaPodataka>
  <tablica1>
    <nTorka>
      <atribut1>...</atribut1>
      ...
      <atributN>...</atributN>
    </nTorka>
    ...
  </tablica1>
  ...
  < tablicaM>
    <nTorka>
      <atribut1>...</atribut1>
      ...
      <atributN>...</atributN>
    </nTorka>
    ...
  </tablicaN>
</bazaPodataka>
```

```
CREATE DATABASE bazaPodataka;
CREATE TABLE tablica1
(  atribut1 ...,
  ...
  atributN ...,
)
...

CREATE TABLE tablicaN
(  atribut1 ...,
  ...
  atributN ...,
)
```

- Korijenski element preslikava se u bazu podataka
- Elementi djeca korijena se preslikavaju u tablicu (tablice)
- Djeca elemenata se preslikavaju u attribute

# Objektno-relacijsko preslikavanje

---

- Dokument se oblikuje kao stablo objekata
- Provodi se u 2 koraka
  - Preslikavanje XML sheme u objekte
  - Preslikavanje objekata u relacijsku shemu
- Kompleksni tipovi podataka se preslikavaju u klase a primitivni u skalarna obilježja klasa
- Nije prikladno za dokumente s mješovitim sadržajem
- Ne dovoditi u vezu sa objektno-relacijskim bazama podataka

# Objektno-relacijsko preslikavanje-primjer

```

<?xml version="1.0" encoding="utf-8"?>
<studAdmin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\shema.xsd">
<upisi>
  <upisGodine>
    <JMBAG>0036123456</JMBAG>
    <akGodina>2008</akGodina>
    <datumUpis>01.07.2008</datumUpis>
    <upisanPred>
      <sifPred>432</sifPred>
      <semestar>1</semestar>
    </upisanPred>
    <upisanPred>
      <sifPred>692</sifPred>
      <semestar>1</semestar>
    </upisanPred>
  </upisGodine>
</upisi>
<predmeti>
  <predmet>
    <sifPred>432</sifPred>
    <nazivPred>Matematika 1</nazivPred>
    <ECTSBod>5.0</ECTSBod>
  </predmet>
  <predmet>
    <sifPred>692</sifPred>
    <nazivPred>Vještine komuniciranja</nazivPred>
    <ECTSBod>2.0</ECTSBod>
  </predmet>
</predmeti>
</studAdmin>

```

```

<studenti>
  <student>
    <JMBAG>0036123457</JMBAG>
    <imeStud>Ivo</imeStud>
    <prezStud>Perić</prezStud>
  </student>
  <student>
    <JMBAG>0036123456</JMBAG>
    <imeStud>Ana</imeStud>
    <prezStud>Marić</prezStud>
  </student>
</studenti>
</studAdmin>

```

# Objektno-relacijsko preslikavanje-primjer

## Relacijska shema

```
CREATE TABLE student
(  JMBAG      CHAR(10) PRIMARY KEY CONSTRAINT pkStud,
   JMBG       CHAR(13) UNIQUE CONSTRAINT uiJMBG,
   imeStud   NCHAR(50),
   prezStud  NCHAR(50),

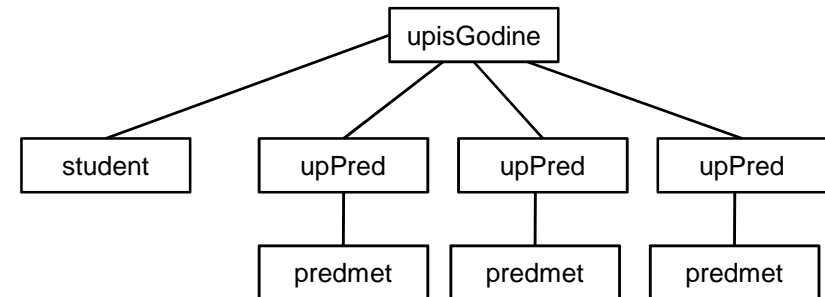
   CHECK JMBAG MATCHES
      '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
)

CREATE TABLE predmet
(  sifPred   INTEGER PRIMARY KEY CONSTRAINT pkPred,
   nazPred  NCHAR(120),
   ECTSBoD  DECIMAL(3,2)
)

CREATE TABLE upisGodine
(  akGodina  SMALLINT,
   JMBAG     CHAR(10) REFERENCES stud(JMBAG) CONSTRAINT fkUpisGodineStud,
   datumUpisa DATE,
   PRIMARY KEY (JMBAG, akGodina) CONSTRAINT pkUpisGodine
)

CREATE TABLE upPred
(  akGodina  SMALLINT,
   JMBAG     CHAR(10),
   sifPred   INTEGER,
   semestar  SMALLINT
   PRIMARY KEY (JMBAG, akGodina, sifPred, semestar) CONSTRAINT pkUpPred,
   FOREIGN KEY (sifPred) REFERENCES pred(sifPred) CONSTRAINT fkUpPredPred
```

## Objekti



# Normalizacija i referencijski integritet (1)

---

## Prva normalna forma

### Relacijska shema

- domene atributa sadrže samo jednostavne (nedjeljive) vrijednosti
- vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa
- neključni atributi relacije funkcijski ovise o ključu relacije

### XML shema

- može sadržavati promjenjiv broj “polja” (usporedivo s null vrijednostima u RDB)
- dozvoljene višestruke vrijednosti za “polja” (maxoccurs) - čvor roditelj može imati više čvorova djece
- tip podatka “polja” ne mora biti striktno zadan nego može biti jedan od predviđenih (choice)
- “polja” mogu biti kompleksnog tipa



# Normalizacija i referencijski integritet (2)

---

## Druga i treća normalna forma

- 2NF podržana  
zbog hijerarhijske strukture nema ponavljanja nadređenih podataka (primjer roditelj–djeca)
- 3NF problem  
razbijanje dokumenata stvara prevelik broj dokumenata s kojima se teško upravlja  
općenito se smatra da ponavljanje podataka nije problem (?)

# Integritetska ograničenja u relacijskoj i u XML shemi

Relacijska shema	XML Shema	DTD
NOT NULL	nillable="false"	REQUIRED
UNIQUE	unique	
PRIMARY KEY	key	ID
FOREIGN KEY	keyref	IDREF

- moguće definirati kompozitni ključ/unique ograničenje
- ključ mogu činiti elementi i/ili atributi
- “referencijski integritet” se realizira pomoću keyref koji se referencira na key
- doseg ključa određen smještajem i hijerarhijom dokumenta

# Integritetska ograničenja u relacijskoj i u XML shemi

## Ključ i unique ograničenje

### Relacijska shema

```
CREATE TABLE student
(JMBAG      CHAR(10) PRIMARY KEY
              CONSTRAINT pkStud,
JMBG       CHAR(13) UNIQUE CONSTRAINT uiJMBG,
imeStud    NCHAR(50) NOT NULL,
prezStud   NCHAR(50) NOT NULL,
CHECK JMBAG MATCHES
' [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
)
```

- vrijednost elementa s imenom JMBAG mora biti poznat a za svaki element student
- vrijednost elementa s imenom JMBAG mora biti jedinstvena za sve elemente student
- vrijednost elementa s imenom JMBG mora biti jedinstvena za sve elemente student – ne mora nužno biti poznata ali samo jedna vrijednost smije biti nepoznata da bi bila jedinstvena

### XML shema

```
<xsd:element name="student">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="JMBAG">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9]{10}" />
          </xsd:restriction>
        </xsd:simpleType>
        <xsd:key name="pkStud">
          <xsd:selector xpath="//student"/>
          <xsd:field xpath="JMBAG"/>
        </xsd:key>
      </xsd:element>
      <xsd:element name="JMBG" type="xsd:string">
        <xsd:unique name="uiJMBG">
          <xsd:selector xpath="//student"/>
          <xsd:field xpath="JMBG"/>
        </xsd:unique>
      </xsd:element>
      <xsd:element name="imeStud" type="xsd:string" nillable="false"/>
      <xsd:element name="prezStud" type="xsd:string" nillable="false"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

# Integritetska ograničenja u relacijskoj i u XML shemi

## Referencijski integritet

### Relacijska shema

```
CREATE TABLE upisGodine
(   akGodina    SMALLINT,
    JMBAG        CHAR(10) REFERENCES stud(JMBAG) CONSTRAINT fkUpisGodineStud,
    datumUpisa  DATE,
    PRIMARY KEY (JMBAG, akGodina) CONSTRAINT pkUpisGodine
)
```

- vrijednost elementa s imenom JMBAG u elementu upisGodine mora odgovarati točno jednoj vrijednosti elementa JMBAG u elementu student

### XML shema

```
<xsd:element name="upisGodine">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="JMBAG" type="xsd:string">
        <xsd:keyref name="fkUpisGodineStud" refer="pkStud">
          <xsd:selector xpath="//upisGodine"/>
          <xsd:field xpath="JMBAG"/>
        </xsd:keyref>
      </xsd:element>
      <xsd:element name="akGodina" type="xsd:integer"/>
      <xsd:element name="datumUpis" type="xsd:date"/>
      <xsd:element ref="upisPred" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

# XML enabled baze podataka

---

- u relacijske sustave ugrađeni mehanizmi koji olakšavaju rad s XML dokumentima
- Ne uvode novi model za rad s XML-om već proširuju postojeći – relacijski.
- XML enabled baza podataka je „svjesna” da radi s XML strukturama i u skladu s time nudi različite servise
- Nad XML podacima implementirane mogućnosti za pretraživanje teksta specifične za XML dokumente
- Uključuju podršku za upitne jezike (najčešće XPath) i API-je (najčešće DOM = Document Object Model) koji su u skladu s preporukama W3C

# XML enabled baze podataka

---

Neke realizacije XML enabled baza podataka su:

- Oracle 9i
- Oracle 10g
- Oracle 11g
- Microsoft SQL Server 2000
- Microsoft SQL Server 2005
- Microsoft Access XP
- IBM DB2

# Izvorne XML baze podataka (NXD) (1)

---

- **Definicija 1.** (Gavin Powell: Beginning XML Database, 2007)  
A **native XML database** is essentially any method of storing XML data as an XML document.
- **Definicija 2.** (<http://www.ibm.com/developerworks/xml/library/x-mxd4.html>)  
A **native XML database** is one that treats XML documents and elements as the fundamental structures rather than tables, records, and fields
- **Definicija 3.** ([XML:DB Initiative](#))  
A native XML database...
  - Defines a (logical) model for an XML document -- as opposed to the data in that document -- and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of such models are the XPath data model, the XML Infoset, and the models implied by the DOM and the events in SAX 1.0.
  - Has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage.
  - Is not required to have any particular underlying physical storage model. For example, it can be built on a relational, hierarchical, or object-oriented database, or use a proprietary storage format such as indexed, compressed files.



## Izvorne XML baze podataka (NXD) (2)

---

- U bazu se može smjestiti samo XML - iz baze se može dobiti samo XML
- Pohranjuju XML dokumente na prirodan način
- Osnovna logička jedinica za pohranu podataka je dokument
  - XML tip podatka je specijalni podatkovni tip
  - ekvivalentna struktura u RDBMS je n-torka
- Održavaju hijerarhiju i značenje XML dokumenta
- Neke XML baze podataka indeksiraju sve, dok druge omogućuju korisnicima da specificiraju polja po kojima će se indeksirati
  - strukturalni – elementi i imena atributa
  - vrijednosni – tekstovi i vrijednosti atributa
  - full text indeksi

# NXD - upitni jezici, izmjene

---

- Upitni jezici
  - mnogi sustavi imaju vlastite jezike
  - u mnogima se koristi XPath
  - XQuery postaje standard
- Izmjene
  - Mnoge baze podataka samo zamjenjuju postojeće dokumente
  - Strategije razvoja upitnih jezika:
    - Izmjene kroz DOM
    - Jezik za izmjene – pomoću XPath se pronađe čvor i nakon toga se obavlja unos (prije ili poslije čvora), izmjena čvora, brisanje čvora
  - Proširenje XQuery

# Transakcije, zaključavanje, paralelni pristup

---

- Mnoge podržavaju transakcije
- Zaključavanje se obično obavlja na razini dokumenta (a ne fragmenta)
  - zaključavanje čvora obično povlači zaključavanje roditelja itd.
    - ako transakcija zahtijeva promjenu podatka u “listu” logično je zaključati njegovog roditelja jer bi ga inače netko u međuvremenu mogao npr. obrisati čime bi obrisao i “list”
    - to ne bi smjelo onemogućiti drugu transakciju da mijenja dijelove dokumenta koji nisu direktno na putu od korijena do lista koje koristi prva transakcija
- Problem paralelnog pristupa ovisi o:
  - sadržaju koji je pohranjen u dokumentima
  - broju istovremenih korisnika
- U budućnosti se očekuje mogućnost zaključavanja fragmenata

# Kada i zašto koristiti izvornu (native) XML bazu podataka?

---

- Kada pohranjujemo nestrukturirane ili polu-strukturirane podatke
  - Pohrana u tablice relacijske baze podataka bi rezultirala prevelikim brojem tablica i/ili prevelikim brojem atributa s mnogo NULL vrijednosti
- Kada se često mijenja shema podataka
  - Za XML dokumente pohranjene u NXD se ne mora (ali može) definirati shema
- Kada je važno sačuvati originalni izgled dokumenata
  - pohranom u relacijsku bazu on se nužno izgubi
- Zato što NXD brže dohvaća XML dokumente
  - Podaci nisu razasuti po tablicama kao kod RDB
  - Obično NXD pohranjuje cijeli XML dokument na fizički kontinuiranom memorijskom prostoru ili koristi fizičke veze (pokazivače) između dijelova dokumenta

# Kada i zašto koristiti izvornu (native) XML bazu podataka?

---

- Zbog upitnog jezika prilagođenog XML tipu podatka
- Zbog brzine dohvata koja koristi indeksne strukture prilagođene XML tipu podatka
- Zbog mogućnosti pretrage cjelokupnog teksta

# XML u datotečnom sustavu - XML u NXD

	<b>Datotečni sustav</b>	<b>NXD</b>
<b>Logički smještaj i imenovanje XML dokumenata</b>	Razasuti po direktorijima. Potrebno voditi brigu o imenovanju dokumenata.	Svi na jednom mjestu – u bazi podataka. Imena dokumenata su nebitna.
<b>Primjenjivost u višekorisničkom okruženju</b>	Ne (teško i u jednokorisničkom)	Da
<b>Zaštita podataka i granulacija razina ovlasti</b>	Nikako ili teško provedivi	Jedna od standardnih mogućnosti SUBP
<b>Različit pogled na iste podatke</b>	Svodi se na mukotrpnu izradu izvještaja	Posljedica drugačije formulacije upita
<b>Brzina dohvata</b>	Spora pretraga dokumenata. Nema indeksa na nivou operacijskog sustava u čiji datotečni sustav su smješteni.	Indeksne strukture prilagođene XML tipu podatka. Za ubrzavanje dohvata se koriste informacije prikupljene o dokumentima prilikom unosa/izmjene.
<b>Upiti nad više dokumenata</b>	Ne	Da

# Najpoznatije realizacije NXD

---

## Najpoznatije realizacije NXD baza podataka

- Otvorenog kôda
  - XIndex (Apache)
  - eXist
  - oZone
  - MonetDB/XQuery
  - myXMLDB
  - Sedna
- Komercijalni
  - Gemfire Enterprise
  - DOMSafeXML
  - Tamino (Software AG)



# Literatura

---

- A.B. Chaundry, A. Rashid, R. Zicari: **XML Data Management Native XML and XML/Enabled Database Systems**, Addison-Wesley, 2003
- G. Powel: **Beginning XML Databases**, Wiley Publishing, Inc, 2007
- K. Williams: **Professional XML Databases**, Wrox Press Ltd, 2000
- <http://www.w3.org/XML>
- <http://www.xml.com>
- <http://www.rpbourret.com/>