



## XML model podataka

---

*XML Schema jezik i XML baze podataka*

# Sadržaj

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

# Motivacija

---

- **Internet aplikacije i aplikacije za elektronsku trgovinu**
  - višeslojne K/S arhitekture
  - nude *web* interfejsse
    - pristup informacijama smeštenim u jednu ili više baza podataka
    - putem *web* stranice
      - specifikacija i formatiranje podataka putem HTML-a
        - » *eng. HyperText Markup Language*
  - uobičajeno zahtevaju značajnu razmenu podataka između različitih sistema
    - npr. B2B, B2C

# Motivacija

---

- **Problem**

- HTML nije pogodan za specifikaciju strukturiranih podataka dobijenih iz BP
  - pogodan je za specifikaciju web dokumenata
  - predefinisani skup tagova
    - nije proširiv od strane korisnika

- **Jedan pristup rešavanju problema**

- ***Extensible Markup Language (XML)***

- danas široko zastupljen u praksi
  - prikazivanje strukturiranih podataka
  - razmenu podataka putem web-a
- nastaje iz jezika **Standard Generalized Markup Language (ISO 8879:1986 SGML)**

# Motivacija

---

- ***Extensible Markup Language (XML)***
  - nastaje 1998. godine
  - specifikacija **strukture i semantike** podataka
    - *XML Schema Language*
      - mogućnost slobodnog kreiranja tagova
      - podržava kreiranje specifikacija na nivou apstrakcije logičkih struktura obeležja
    - *XML Language*
      - specificira logičke strukture podataka
        - » instance logičkih struktura obeležja, opisanih putem XML Schema specifikacija

# Motivacija

---

- ***Extensible Markup Language (XML)***
  - specifikacija **prezentacionog formata XML** struktura
    - *Extensible Stylesheet Language (XSL)*
    - *XSL Formatting Objects (XSL-FO)*
  - mogućnost automatskih **transformacija XML** struktura
    - *XSL Transformations (XSLT)*

# Sadržaj

---

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

# XML Model podataka

---

- **Tipovi podataka prema strukturi**
  - strukturirani podaci
  - nestrukturirani podaci
  - polustrukturirani podaci



# XML Model podataka

---

- **Tipovi podataka prema strukturi**
  - **strukturirani podaci**
    - **postoji šema** koja definiše format podataka
    - striktno zadovoljavaju predefinisani format
      - u praksi, očekuje se da takva šema bude razrađena do potrebnog stepena detaljnosti
    - primer
      - podaci u relacionoj bazi podataka
        - » sve torke imaju isti format

# XML Model podataka

---

- **Tipovi podataka prema strukturi**
  - **nestrukturirani podaci**
    - (A) može postojati **šema** koja definiše format podataka
      - tipovi podataka su, međutim, često "preširoki"
        - » nedovoljno detaljni sa stanovišta semantike podataka
    - (B) ne postoji ni šema koja definiše format podataka
      - ne postoji nikakav ugrađeni format kojeg podaci prate
        - » moguće je da postoji nekakav eksterni format koji opisuje semantiku podataka, ali on ne pripada strukturi
  - **primeri**
    - tekstualni ili multimedijalni dokumenti
      - podaci su nestrukturirani sa stanovišta sistema za njihovo skladištenje
    - HTML dokument
      - tagovi određuju formatiranje ali ne i značenje

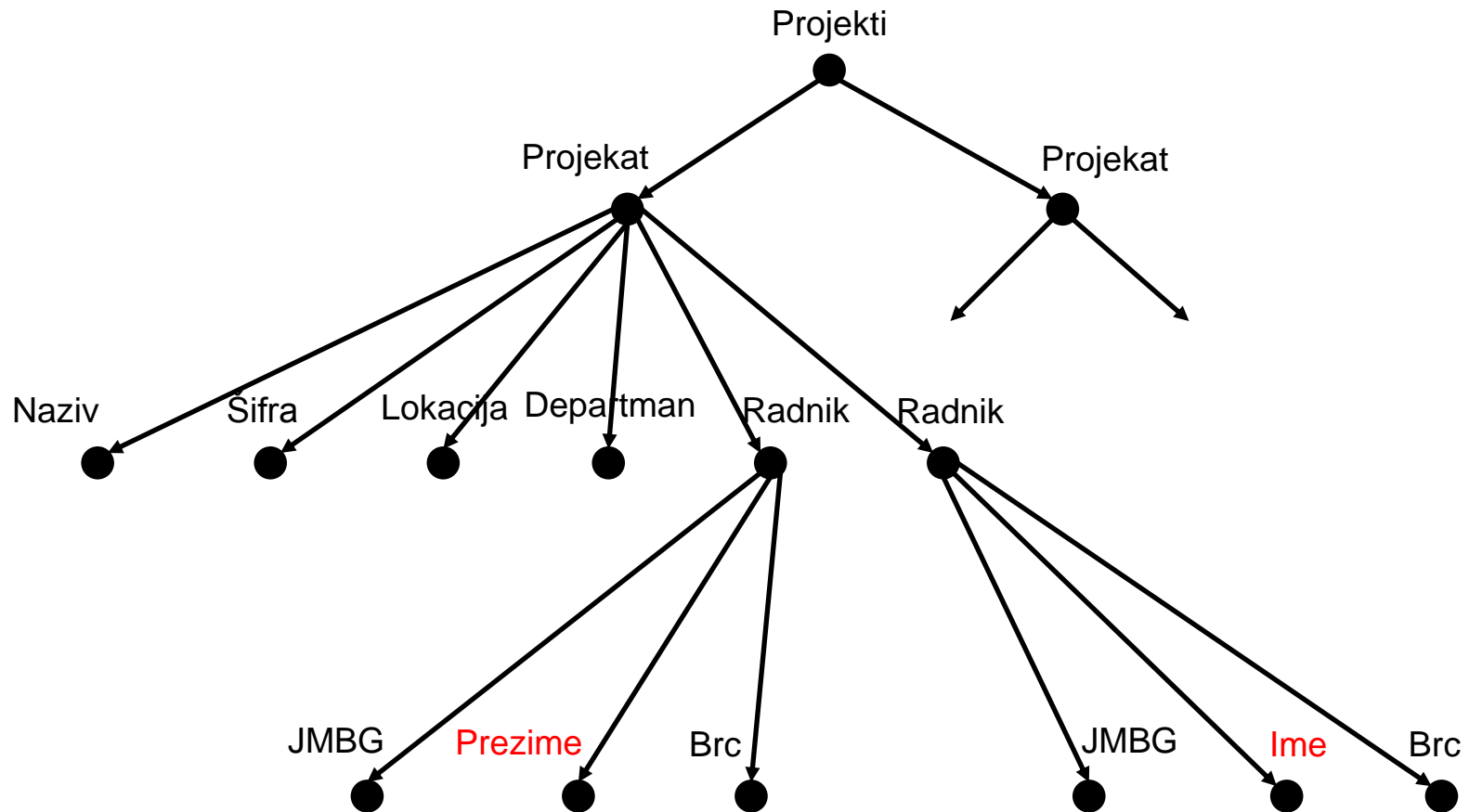
# XML Model podataka

---

- **Tipovi podataka prema strukturi**
  - **polustrukturirani podaci**
    - imaju definisanu strukturu ali postoje i odstupanja
      - atributi mogu da postoje
        - » u svim entitetima
        - » samo u nekim entitetima
    - prikazuju se, često, putem grafova i stabala
  - moguće je definisati šemu
    - koja specificira moguće elemente koji učestvuju u strukturi
      - mogu ali ne moraju svi da postoje

# XML Model podataka

- Primer polustrukturiranih podataka



# XML Model podataka

---

- **XML dokument**

- centralni koncept XML jezika
- konstruiše se pomoću dva osnovna koncepta
  - **element**
    - glavni gradivni element XML dokumenta
    - opisuje se pomoću početnog i krajnjeg taga (oznake)
    - moguće napraviti hijerarhiju elemenata
  - **atribut**
    - karakteristika koja detaljnije opisuje element i reprezentuje njegovu semantiku
    - upotreba atributa ne mora uvek odgovarati upotrebi atributa u relacionom modelu podataka
      - » u XML dokumentima za opis podataka koji se čuvaju u elementima koriste se atributi

# XML Model podataka

---

- **Primer**

- projekti kompanije
- svaki projekat sadrži
  - naziv projekta
  - šifru projekta
  - lokaciju kompanije
  - šifru departmana
  - radnike koji rade na projektu
- svaki radnik sadrži
  - jedinstveni matični broj građanina
  - ime i/ili prezime radnika
  - broj radnih časova na projektu

# XML Model podataka

```
<?xml version= "1.0" standalone="yes"?>
<Projekti>
  <Projekat>
    <Naziv>ProjekatX</Naziv>
    <Sifra>1</Sifra>
    <Lokacija>Novi Sad</Lokacija>
    <Departmentman>5</Departmentman>
    <Radnik>
      <JMBG>123456789</JMBG>
      <Prezime>Petrovic</Prezime>
      <Brc>32.5</Brc>
    </Radnik>
    <Radnik>
      <JMBG>453453453</JMBG>
      <Ime>Petar</Ime>
      <Brc>20.0</Brc>
    </Radnik>
  </Projekat >
  ...
</Projekti>
```

# XML Model podataka

---

- **Element**

- **složeni element**

- konstruisan pomoću drugih elemenata
      - hijerarhija elemenata

- **primitivni (prosti) element**

- sadrži atomične vrednosti

- glavne razlike između XML-a i HTML-a

- nazivi elemenata

- u XML dokumentu reprezentuju semantiku

- u HTML dokumentu definišu isključivo način prezentacije podataka

- specifikacija elemenata

- sadržana je u XML šema dokumentu i korisnički definisana

- u HTML jeziku, svi elementi su unapred definisani



# XML Model podataka

---

- **Atribut**

- najčešće se koristi za **opis podataka** smeštenih u elementima
  - slično atributima u HTML-u
  - za razliku od atributa u bazama podataka koji sadrže same vrednosti
- može da ima istu ulogu kao i atribut u BP
  - kada nije od velikog značaja da XML dokumenti budu čitljivi od strane ljudi

# XML Model podataka

---

- **Atribut**

- može kao vrednosti da sadrži osnovne ili dodatne podatke koje ne sadrži sam element
- tri pristupa strukturiranja podataka
  - svi podaci se pridružuju direktno elementu
  - svi podaci se pridružuju isključivo atributima elementa
  - neki ("osnovni") podaci se pridružuju elementu, a ostali ("dodatni") atributima elementa
- različiti kriterijumi mogu uticati na izbor pristupa koji se može smatrati odgovarajućim u datoj situaciji
  - u literaturi, treći ("hibridni") pristup često se ne preporučuje
  - u prvom pristupu, izuzetak je da se reference ka drugim elementima XML dokumenta iskazuju putem atributa

# XML Model podataka

---

- **Tipovi XML dokumenata**
  - XML dokumenti za opis podataka
  - XML dokumenti za opis realnih dokumenata
  - Hibridni XML dokumenti

# XML Model podataka

---

- **Tipovi XML dokumenata**
  - tip dokumenta nije uvek lako prepoznati
  - definisanjem tipa dokumenata definiše se način skladištenja podataka

# XML Model podataka

---

- **XML dokumenti za opis podataka**
  - *eng. data-centric XML documents*
  - koriste se za razmenu podataka
    - namenjeni pretežno računarskoj obradi
  - visoka granulacija elemenata, čije vrednosti su
    - preuzete iz baze podataka
    - dobijene iz spoljašnjeg izvora
      - npr. merenja ili eksperimenti
  - **strukturirani ili polustrukturirani dokumenti**
    - u zavisnosti od postojanja XML šeme
  - najčešće se čuvaju u relacionim bazama podataka

# XML Model podataka

---

- **Primer**

```
<Letovi>
  <Prevoznik>ABC Airways</Prevoznik>
  <Mesto_polaska>Beograd</Mesto_polaska>
  <Odrediste>Sarajevo</Odrediste>
  <Let>
    <Polazak>09:15</Polazak>
    <Dolazak>10:00</Dolazak>
  </Let>
  <Let>
    <Polazak>11:15</Polazak>
    <Dolazak>12:00</Dolazak>
  </Let>
  <Let>
    <Polazak>13:15</Polazak>
    <Dolazak>14:00</Dolazak>
  </Let>
</Letovi>
```

# XML Model podataka

---

- **XML dokumenti za opis podataka**
  - mogu obuhvatati i manje granulirane elemente
    - ne ide se uvek do nivoa atomičnih podataka
  - mogu sadržati i tekst izvan elemenata
    - ista struktura sadržaja u svim dokumentima
    - mogu se napraviti iz drugih XML dokumenata koji opisuju podatke

# XML Model podataka

---

- **Primer**

<Letovi\_informacije >

<Prevoznik>ABC Airways</Prevoznik> nudi <Broj>tri</Broj> leta dnevno od <Mesto\_polaska>Beograd</Mesto\_polaska> do <Destination>Sarajevo</Destination>. Vremena polaska su <Polazak>09:15</Polazak>, <Polazak>11:15</Polazak>, i <Polazak>13:15</Polazak>. Dolasci se očekuju sat vremena kasnije.

</Letovi\_informacije>



# XML Model podataka

---

- **XML dokumenti za opis realnih dokumenata**
  - *eng. document-centric XML documents*
  - sadrže veliku količinu teksta
    - namenjeni ljudima
    - npr. radovi ili knjige
  - najčešće **nestrukturirani** dokumenti
    - većina nestrukturiranih elemenata
      - niska granulacija
  - obično nastaju izvan baze podataka
    - npr. pdf, rtf, ili sgml dokumenti
      - koji su konvertovani u XML
  - čuvaju se u XML bazama podataka

# XML Model podataka

- **Primer**

<Proizvod>

<Uvod>

The <Naziv\_proizvoda>Turkey Wrench</ Naziv\_proizvoda > from <Proizvodjac>Full Fabrication Labs, Inc.</ Proizvodjac> is <Sazetak>like a monkey wrench, but not as big.</Sazetak>

</Uvod>

<Opis>

<Para>The turkey wrench, which comes in <i>both right- and left-handed versions (skyhook optional)</i>, is made of the <b>finest stainless steel</b>. The Readi-grip rubberized handle quickly adapts to your hands, even in the greasiest situations. Adjustment is possible through a variety of custom dials.</Para>

</Opis>

...

</Proizvod>

# XML Model podataka

---

- **Hibridni XML dokumenti**
  - *eng. hybrid-centric XML documents*
  - jedan deo dokumenta je strogo strukturiran, dok je drugi slabo strukturiran ili potpuno nestructuriran
  - mogu biti opisani XML šemom

# XML Model podataka

---

- **XML Model podataka**
  - šema baze podataka predstavlja model tipa stabla
  - analogija s hijerarhijskim modelom podataka
- **Struktura XML šeme BP**
  - listovi predstavljaju proste elemente
  - neterminalni čvorovi su složeni elementi
    - ne postoji ograničenje dubine hijerarhije

# Sadržaj

---

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

# DTD i XML Schema

---

- **Dobro formiran XML dokument**
  - počinje XML deklaracijom
  - zadovoljava sintaksna ograničenja modela podataka
    - jedan korenski element
    - svaki element mora imati otvarajući i zatvarajući tag

# DTD i XML Schema

---

- **Dobro formiran XML dokument**
  - može biti procesiran od strane generičkog procesora XML dokumenata
    - obezbeđuje navigaciju po stablu elemenata
      - parsiranje obavlja *Simple API for XML* (SAX)
        - » prilikom susretanja svakog otvarajućeg i zatvarajućeg taga SAX obaveštava program za procesiranje dokumenta
          - » *eng. callback*
  - kreira internu reprezentaciju stabla u memoriji
    - najčešće u formi tzv. *Document Object Model*-a (DOM)

# DTD i XML Schema

---

- **Dobro formiran XML dokument**
  - ne zahteva postojanje šeme
    - ne postoji predefinisani skup elemenata koji se mogu pojaviti u dokumentu
    - dozvoljava autoru dokumenta širu slobodu u definisanju elemenata
    - onemogućava automatsku interpretaciju značenja podataka u dokumentu



# DTD i XML Schema

---

- **Validan XML dokument**
  - dobro formiran XML dokument
  - koji zadovoljava šemu
    - struktura elemenata u XML dokumentu mora da prati specifikaciju strukture definisane u posebnom dokumentu
  - jezici za specifikaciju strukture XML dokumenta
    - XML *Document Type Definition* (DTD)
    - XML *Schema*

# DTD i XML Schema

---

- ***Document Type Definition (DTD)***
  - deo osnovnog XML standarda
  - obično se čuva odvojeno od XML dokumenata
    - poseban dokument sa ekstenzijom *.dtd*
  - opisuje format XML dokumenta
    - definiše elemente i attribute
      - njihova imena i tagove
    - definiše tip sadržaja elemenata i atributa

# DTD i XML Schema

- **Primer**

```
<!DOCTYPE Projekti [  
  <!ELEMENT Projekti (Projekat+)>  
  <!ELEMENT Projekat (Naziv, Sifra, Lokacija, Departman?, Radnici)  
    <!ATTLIST Projekat  
      ProjId ID #REQUIRED>  
  >  
  <!ELEMENT Naziv (#PCDATA)>  
  <!ELEMENT Sifra (#PCDATA)>  
  <!ELEMENT Lokacija (#PCDATA)>  
  <!ELEMENT Departman (#PCDATA)>  
  <!ELEMENT Radnici (Radnik*)>  
  <!ELEMENT Radnik (JMBG, Prezime?, Ime?, Brc)>  
  <!ELEMENT JMBG (#PCDATA)>  
  <!ELEMENT Prezime (#PCDATA)>  
  <!ELEMENT Ime (#PCDATA)>  
  <!ELEMENT Brc (#PCDATA)>  
>
```

# DTD i XML Schema

---

- **DTD dokument**

- provera validnosti XML dokumenta vrši se u odnosu na DTD dokument
  - ime DTD dokumenta se navodi u zaglavlju
  - *standalone* atribut definiše da li se dokument proverava u odnosu DTD dokument
    - ukoliko je njegova vrednost „no“ dokument se proverava

```
<?xml version="1.0" standalone="no"?>  
<!DOCTYPE Projekti SYSTEM "projekti.dtd">
```

# DTD i XML Schema

---

- **Mane DTD-a**
  - skromne mogućnosti u definisanju tipova podataka
    - samo 10 tipova podataka
  - sintaksa drugačija od XML-a
    - zahteva postojanje posebnih procesora dokumenata
  - svi XML dokumenti moraju da poseduju elemente baš u onom redosledu u kom su definisani u DTD dokumentu
    - nije moguće specificirati broj elemenata bez specifikacije njihovog redosleda
  - nije moguća specifikacija ograničenja vezanih za skladištenje podataka u BP

# DTD i XML Schema

---

- **XML Schema**

- jezik za opis strukture XML dokumenata
  - de facto standard
  - poštuje ista sintaksna pravila kao XML
    - može se koristiti isti procesor dokumenata
- uvođenjem XML Schema jezika imamo dve vrste dokumenata
  - XML Schema dokument
    - opis strukture
  - XML dokument
    - sadrži vrednosti
    - struktura zadovoljava pravila navedena u XML schema dokumentu
      - » ukoliko on postoji

# DTD i XML Schema

---

- **XML Schema**
  - zasnovan na modelu tipa stabla
    - elementi i atributi su glavni strukturalni koncepti
  - dodatni koncepti preuzeti su iz relacionog i objektnog modela
    - ključevi, reference i identifikatori

# DTD i XML Schema

---

- **XML Schema**
  - XML *Schema* dokument može da obuhvata
    - opis šeme i XML prostore imena
    - anotacije
    - elemente šeme
    - tipove elemenata
    - integritet šeme



# DTD i XML Schema

---

- **Opis XML šeme i XML prostori imena**
  - specifikacija elemenata za opis XML šeme
    - svakom specifikacijom korišćene šeme specificira se **XML prostor imena**
      - sprečava koliziju elemenata sa istim imenom
        - » definiše skup elemenata koji imaju isti prefiks
      - jedan tip dokumenta može uključivati elemente iz više prostora imena
      - jedan element može biti korišćen u više tipova dokumenata
    - svakom prostoru imena dodeljuje se skraćeno ime
      - koristi se kao prefiks za elemente iz te šeme

# DTD i XML Schema

---

- Opis XML šeme i XML prostori imena

```
<?xml version="1.0" encoding="UTF-8" ?>  
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
...  
</xsd:schema>
```

# DTD i XML Schema

---

- **Anotacija**

- specifikacija komentara i opisa *XML schema* dokumenta
  - *xsd:documentation*
    - specifikacija komentara namenjenog ljudima
    - atribut *xml:lang* specificira korišćeni jezik u samom XML dokumentu
  - *xsd:appinfo*
    - specifikacija komentara namenjenog aplikacijama

# DTD i XML Schema

---

- Anotacija

```
<xsd:schema ... >
```

```
...
```

```
  <xsd:annotation>
```

```
    <xsd:documentation xml:lang="sr">Šema kompanije kreirana za  
    skladištenje podataka u XML bazi podataka</xsd:documentation>
```

```
  </xsd:annotation>
```

```
...
```

```
</xsd:schema>
```

# DTD i XML Schema

---

- **Element šeme**

- služi za definisanje elementa koji može biti korišćen u XML dokumentima
  - jedan **korenski element**
  - atribut *name* specificira ime elementa u XML dokumentu
  - između otvarajućeg i zatvarajućeg taga navodi se struktura elementa
- **elementi prvog reda** su elementi koji se nalaze neposredno ispod korenskog elementa u hijerarhiji elemenata

# DTD i XML Schema

---

- **Tip podatka u elementu**
  - definiše se putem *type* atributa
    - ukoliko je navedena vrednost
      - struktura elementa definiše se nezavisno od samog elementa
    - ukoliko nije navedena vrednost
      - struktura elementa navodi se u okviru samog elementa
      - obično *xsd:complexType*
        - » složeni element
  - atributi *minOccurs* i *maxOccurs*
    - definišu minimalan i maksimalan broj ponavljanja datog elementa
    - podrazumevana vrednost je jedan

# DTD i XML Schema

---

- **Kompleksni element**

- definiše se putem *xsd:complexType* elementa
  - sadrži podelemente
    - uređena ili neuređena lista elemenata
    - navodi se broj mogućih ponavljanja svakog elementa

- **Kompleksni atributi**

- takođe se definišu putem *xsd:complexType* elementa

# DTD i XML Schema

```
<xsd:element name="projekti">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="departman" type="Departman"
        minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="radnik" type="Radnik" minOccurs="0"
        maxOccurs="unbounded">
        <xsd:unique name="izdržavanilmeUnique">
          <xsd:selector xpath="radnikIzdrzava" />
          <xsd:field xpath="izrzavanilme" />
        </xsd:unique>
      </xsd:element>
      <xsd:element name="projekat" type="Projekat" minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType> ...
</xsd:element/>
```



# DTD i XML Schema

---

```
<xsd:complexType name="Department">
  <xsd:sequence>
    <xsd:element name="departmentNaziv" type="xsd:string" />
    <xsd:element name="departmentSifra" type="xsd:string" />
    <xsd:element name="departmentRukJMBG" type="xsd:string" />
    <xsd:element name="departmentRukDatumPocetka" type="xsd:date" />
    <xsd:element name="departmentLokacija" type="xsd:string"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

# DTD i XML Schema

```
<xsd:complexType name="Radnik">
  <xsd:sequence>
    <xsd:element name="radnikIme" type="Ime" />
    <xsd:element name="radnikJMBG" type="xsd:string" />
    <xsd:element name="radnikPol" type="xsd:string" />
    <xsd:element name="radnikPlata" type="xsd:unsignedInt" />
    <xsd:element name="radnikDatumRodjenja" type="xsd:date" />
    <xsd:element name="radnikDepartmanSifra" type="xsd:string" />
    <xsd:element name="radnikSefJMBG" type="xsd:string" />
    <xsd:element name="radnikAdresa" type="Adresa" />
    <xsd:element name="radnikRadiNa" type="RadiNa" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:element name="radnikIzdrzava" type="Izdrzavani"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

# DTD i XML Schema

---

```
<xsd:complexType name="Projekat">
  <xsd:sequence>
    <xsd:element name="projekatNaziv" type="xsd:string" />
    <xsd:element name="projekatSifra" type="xsd:string" />
    <xsd:element name="projekatLokacija" type="xsd:string" />
    <xsd:element name="projekatDepartmentmanSifra" type="xsd:string" />
    <xsd:element name="projekatRadProj" type="RadProj" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

# DTD i XML Schema

```
<xsd:complexType name="Izdrzavani">
  <xsd:sequence>
    <xsd:element name="izdrzavanilme" type="xsd:string" />
    <xsd:element name="izdrzavaniPol" type="xsd:string" />
    <xsd:element name="izdrzavaniDatumRodjenja" type="xsd:date" />
    <xsd:element name="izdrzavaniOdnosSaRadnikom" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="Adresa">
  <xsd:sequence>
    <xsd:element name="broj" type="xsd:string" />
    <xsd:element name="ulica" type="xsd:string" />
    <xsd:element name="grad" type="xsd:string" />
    <xsd:element name="drzava" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

# DTD i XML Schema

---

```
<xsd:complexType name="Ime">  
  <xsd:sequence>  
    <xsd:element name="ime" type="xsd:string" />  
    <xsd:element name="srednjelme" type="xsd:string" />  
    <xsd:element name="prezime" type="xsd:string" />  
  </xsd:sequence>  
</xsd:complexType>
```

# DTD i XML Schema

---

```
<xsd:complexType name="RadProj">  
  <xsd:sequence>  
    <xsd:element name="radnikJMBG" type="xsd:string" />  
    <xsd:element name="brc" type="xsd:float" />  
  </xsd:sequence>  
</xsd:complexType>
```

```
<xsd:complexType name="RadiNa">  
  <xsd:sequence>  
    <xsd:element name="projekatSifra" type="xsd:string" />  
    <xsd:element name="brc" type="xsd:float" />  
  </xsd:sequence>  
</xsd:complexType>
```

# DTD i XML Schema

---

- **Integritet podataka**

- moguće je definisati

- **ograničenje jedinstvenosti**

- jedinstvena vrednost elementa

- » na nivou XML dokumenta

- može imati nedostajuću (*null*) vrednost

- *xsd:unique* element šeme

- » definiše elemente koji imaju jedinstvene vrednosti

- » podelement: *xsd:selector*

- » definiše tip elementa na koji se primenjuje ograničenje

- » time, definiše opseg ograničenja

- » podelement: *xsd:field*

- » polje unutar tipa na koje se primenjuje ograničenje

- » polje čija je vrednost jedinstvena

# DTD i XML Schema

---

- **Integritet podataka**

- moguće je definisati

- **primarni ključ**

- uvek postoji

- » minimalni kardinalitet je veći od 0

- » *minOccurs* > 0

- ne može imati nedostajuću (*null*) vrednost

- » *nillable*="false"

- element *xsd:key*

- » sadrži podelemente *xsd:selector* i *xsd:field*



# DTD i XML Schema

---

- **Integritet podataka**

- moguće je definisati

- **oganičenja referencijalnog integriteta**

- referencira prethodno definisani primarni ključ

- » broj i tip polja u referenci na ključ mora odgovarati broju i tipu polja u ključu

- element *xsd:keyref*

- » atribut *refer*

- » vrednost je ime primarnog ključa kojeg referenciramo

- » *xsd:selector*

- » referencirajući tip

- » *xsd:field*

- » polje koje predstavlja strani ključ

# DTD i XML Schema

---

```
<xsd:unique name="departmanNazivUnique">  
  <xsd:selector xpath="departman" />  
  <xsd:field xpath="departmanNaziv" />  
</xsd:unique>
```

```
<xsd:unique name="projekatNazivUnique">  
  <xsd:selector xpath="projekat" />  
  <xsd:field xpath="projekatNaziv" />  
</xsd:unique>
```

# DTD i XML Schema

---

```
<xsd:key name="projekatSifraKey">  
  <xsd:selector xpath="projekat" />  
  <xsd:field xpath="projekatSifra" />  
</xsd:key>
```

```
<xsd:key name="departmanSifraKey">  
  <xsd:selector xpath="departman" />  
  <xsd:field xpath="departmanSifra" />  
</xsd:key>
```

```
<xsd:key name="radnikJMBGKey">  
  <xsd:selector xpath="radik" />  
  <xsd:field xpath="radnikJMBG" />  
</xsd:key>
```

# DTD i XML Schema

---

```
<xsd:keyref name="departmanRukJMBGKeyRef"
  refer="radnikJMBGKey">
  <xsd:selector xpath="departman" />
  <xsd:field xpath="departmanRukJMBG" />
</xsd:keyref>
```

```
<xsd:keyref name="radnikDepartmanSifraKeyRef"
  refer="departmanSifraKey">
  <xsd:selector xpath="radnik" />
  <xsd:field xpath="radnikDepartmanSifra" />
</xsd:keyref>
```

```
<xsd:keyref name="radnikSefJMBGKeyRef"
  refer="radnikJMBGKey">
  <xsd:selector xpath="radnik" />
  <xsd:field xpath="radnikSefJMBG" />
</xsd:keyref>
```

# DTD i XML Schema

---

```
<xsd:keyref name="projekatDepartmanSifraKeyRef"
  refer="departmanSifraKey">
  <xsd:selector xpath="projekat" />
  <xsd:field xpath="projekatDepartmanSifra" />
</xsd:keyref>
```

```
<xsd:keyref name="projekatRadProjJMBGKeyRef"
  refer="radnikJMBGKey">
  <xsd:selector xpath="projekat/projekatRadProj" />
  <xsd:field xpath="JMBG" />
</xsd:keyref>
```

```
<xsd:keyref name="radnikRadiNaProjektuSifraKeyRef"
  refer="projekatSifraKey">
  <xsd:selector xpath="radnik/radnikRadiNa" />
  <xsd:field xpath="projekatSifra" />
</xsd:keyref>
```

# Sadržaj

---

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

# XML baze podataka

---

- **Pristupi skladištenju XML dokumenata**
  - skladištenje XML dokumenata u tekstualnom obliku
  - skladištenje sadržaja XML elemenata u bazama podataka
  - skladištenje XML dokumenata u specijalizovanim XML bazama podataka

# Sadržaj

---

- Skladištenje XML dokumenata u tekstualnom obliku
- Skladištenje XML dokumenata u specijalizovanim XML BP
- Skladištenje sadržaja XML elemenata u specijalizovanim XML BP



# XML baze podataka

---

- **Skladištenje XML dokumenata u tekstualnom obliku**
  - ovaj pristup koristi se
    - kada SUBP ima poseban modul za procesiranje dokumenata
    - kada XML dokument nema definisanu šemu
    - za XML dokumente za opis realnih dokumenata
  - XML dokument se skladišti u tekstualnom ili BLOB polju
    - u relacionim, objektno-relacionim i objektnim bazama podataka

# XML baze podataka

---

- **Skladištenje sadržaja XML elemenata u bazama podataka**
  - ovaj pristup koristi se
    - za XML dokumente koji imaju definisanu šemu
  - svi dokumenti imaju istu strukturu
    - predstavljaju strukturirane podatke
  - kreiranje relacione ili objektne šeme baze podataka
    - zahteva specifikaciju mapiranja između šeme baze podataka i šeme XML dokumenta
      - u internom modulu SUBP-a
      - u posebnom međusloju između baze podataka i aplikacije
    - podaci iz elemenata XML dokumenta se skladište u obliku torki ili objekata

# XML baze podataka

---

- **Mapiranje XML šeme na relacionu šemu baze podataka**
  - mapiranje obuhvata
    - elemente
    - attribute
    - tekst
  - ignorišu se
    - entiteti
    - CDATA sekcije
    - komentari
    - procesne instrukcije
    - redosled elemenata

# XML baze podataka

---

- **Mapiranje XML šeme na relacionu šemu baze podataka**
  - snimanje sadržaja dokumenta u BP i njegovo ponovno čitanje iz BP u opštem slučaju **neće dati isti dokument**
    - usled ignorisanja pojedinih delova dokumenta prilikom mapiranja
  - tipovi mapiranja
    - **mapiranje zasnovano na tabelama**
    - **objektno-relaciono mapiranje**

# XML baze podataka

- **Mapiranje XML dokumenata zasnovano na tabelama**
  - struktura XML dokumenata mora imati oblik

```
<database>
  <table>
    <row>
      <column1>...</column1>
      <column2>...</column2>
      ...
    </row>
    <row> ... </row>
    ...
  </table>
  <table> ... </table>
  ...
</database>
```

# XML baze podataka

---

- **Mapiranje XML dokumenata zasnovano na tabelama**
  - u slučaju jedne tabele struktura XML dokumenta ima oblik

```
<table>  
  <row>  
    <column1>...</column1>  
    <column2>...</column2>  
    ...  
  </row>  
  <row> ... </row>  
  ...  
</table>
```

# XML baze podataka

---

- **Mapiranje XML dokumenata zasnovano na tabelama**
  - moguće je definisati da li se kolona iz relacije BP mapira na element ili na atribut
  - meta-podaci mogu biti uključeni u zaglavlje dokumenta ili kao atributi elemenata tabele ili kolone

# XML baze podataka

---

- **Mapiranje XML dokumenata zasnovano na tabelama**
  - pojam tabele u mapiranju
    - nije nužno ekvivalentan pojmu tabele u relacionoj BP
    - BP -> XML
      - pojam tabele obuhvata bilo koji rezultujući skup
        - » dobijen kao rezultat upita
    - XML -> BP
      - pojam tabele obuhvata tabelu ili pogled baze podataka



# XML baze podataka

---

- **Mapiranje XML dokumenata zasnovano na tabelama**
  - koristi se prilikom serijalizacije podataka iz relacione baze podataka
    - prilikom transfera podataka između dva RSUBP-a
  - koriste ga pojedine aplikacije srednjeg sloja za razmenu podataka između XML dokumenata i relacione baze podataka
  - mana ovog mapiranja
    - XML dokumenti **moraju da zadovoljavaju prethodno definisani format**

# XML baze podataka

---

- **Objektno-relaciono mapiranje XML dokumenata**
  - postoje dve faze
    - XML model -> objektni model
    - objektni model -> relacioni model

# XML baze podataka

---

- **Objektno-relaciono mapiranje XML dokumenata**
  - XML model -> objektni model
    - podaci iz XML dokumenta se interno u mapperima predstavljaju kao stablo objekata
    - složeni tipovi u XML dokumentima predstavljaju se pomoću **klasa**
    - prosti tipovi u XML dokumentima predstavljaju se pomoću **skalara**
  - objektni model dobijen prilikom mapiranja **nije predstavljen** pomoću DOM-a
    - DOM ima istu strukturu za svaki XML dokument
    - objektni model dobijen mapiranjem razlikuje se za dokumente sa različitom šemom

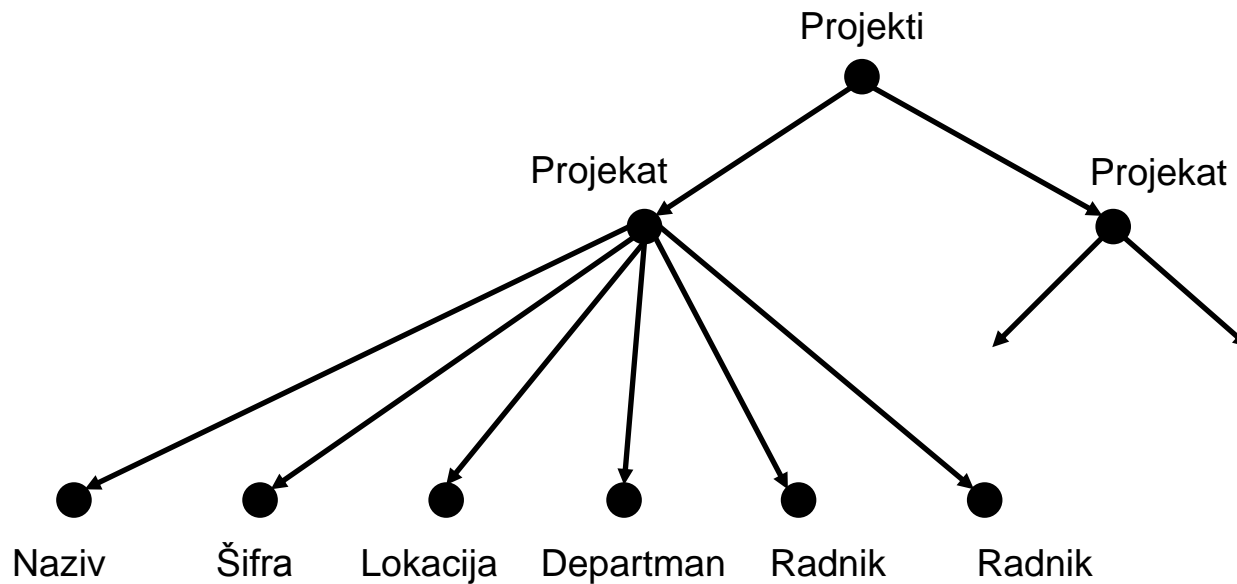
# XML baze podataka

---

- **Objektno-relaciono mapiranje XML dokumenata**
  - objektni model -> relacioni model
    - mapira se pomoću tradicionalnih objektno/relacionih mapera
      - klase -> tabele
      - skalari -> kolone
      - kompozitni objekti -> strani ključevi ka drugim tabelama

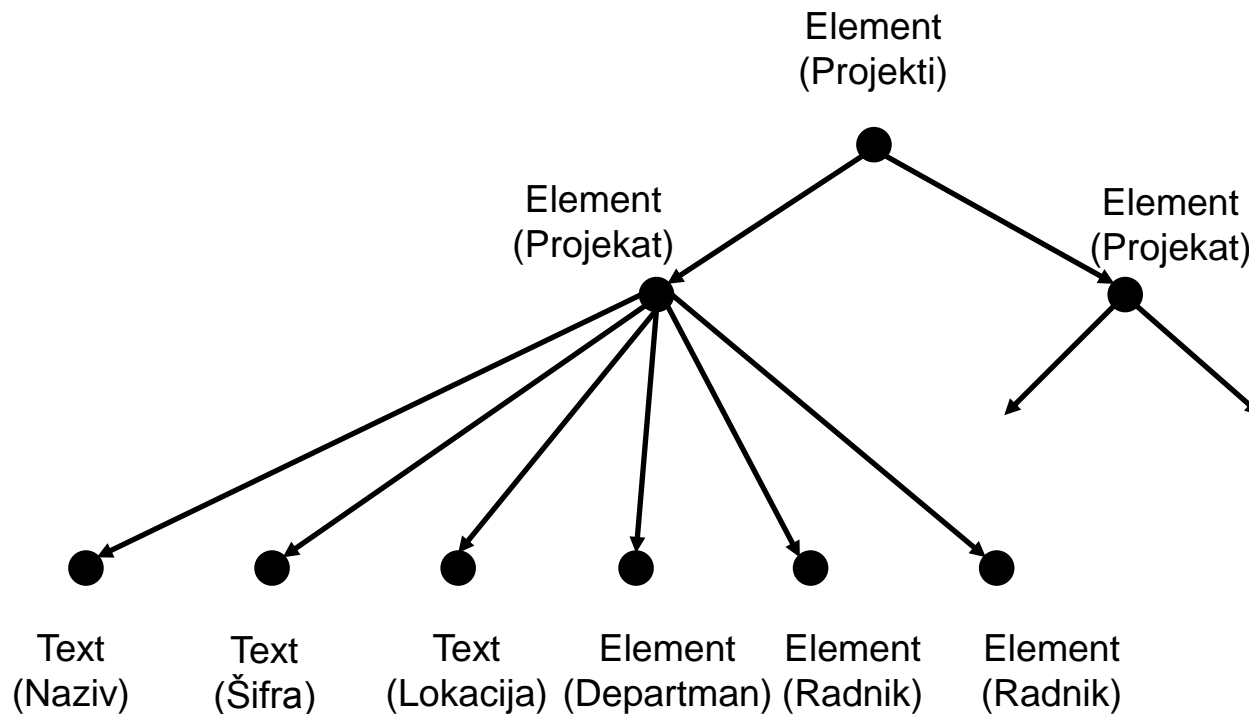
# XML Model podataka

- Primer modela XML dokumenta



# XML Model podataka

- Primer DOM modela XML dokumenta



# XML Model podataka

- **Primer mapiranja XML elemenata na klase**

```
<xsd:complexType name="Projekat">
  <xsd:sequence>
    <xsd:element name="projekatNaziv" type="xsd:string" />
    <xsd:element name="projekatSifra" type="xsd:string" />
    <xsd:element name="projekatLokacija" type="xsd:string" />
    <xsd:element name="projekatDepartmanSifra" type="xsd:string" />
    <xsd:element name="projekatRadProj" type="RadProj"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

# XML Model podataka

---

- **Primer mapiranja XML elemenata na klase**

```
class Projekat {  
    String projekatNaziv;  
    String projekatSifra;  
    String projekatLokacija;  
    String projekatDepartmanSifra;  
    RadProj [ ] projekatRadProj;  
}
```



# Sadržaj

---

- Skladištenje XML dokumenata u tekstualnom obliku
- Skladištenje XML dokumenata u specijalizovanim XML BP
- Skladištenje sadržaja XML elemenata u specijalizovanim XML BP

# XML baze podataka

---

- **Skladištenje XML dokumenata u specijalizovanim XML bazama podataka**
  - XML baze podataka
    - *eng. Native XML databases*
    - namenjene isključivo za skladištenje XML dokumenata
    - zasnovane na XML modelu podataka
    - u potpunosti čuvaju strukturu dokumenata
    - mogu skladištiti
      - sadržaj XML dokumenata
        - » retko se koristi ovaj tip skladištenja
        - » obično se takvi dokumenti mapiraju na relacione BP
      - XML dokumente
        - » često skladišteni u XML bazi podataka

# XML baze podataka

---

- **XML SUBP**
  - definiše logički model XML dokumenta
    - dokumentima rukuje prema tom modelu
    - model obuhvata elemente, attribute, tekst i redosled
    - primeri ovakvih modela: XPath model, XML infoset itd.
  - XML dokument je osnovna logička jedinica skladištenja
    - u relacionim bazama to je toraka u tabeli
    - upiti ili ažuriranje mogu da obuhvate samo deo dokumenta

# XML baze podataka

---

- **XML SUBP**
  - ne mora da poseduje sopstveni (specifični) model za fizičko skladištenje podataka
    - može se oslanjati na relacionu, hijerarhijsku, OO bazu, ili na indeksirane datoteke

# XML baze podataka

---

- **XML SUBP podržavaju**
  - kolekcije dokumenata
  - upitne jezike
  - ažuriranje sadržaja
  - transakcije, zaključavanje i konkurentno izvršavanje
  - programski pristup (API)
  - očuvanje izvorne strukture dokumenta
  - skladištenje podataka van baze podataka
  - indeksiranje
  - skladištenje spoljašnjih entiteta
  - normalizaciju
  - ograničenje referencijalnog integriteta

# XML baze podataka

---

- **XML SUBP**
  - **kolekcije dokumenata**
    - podržava grupisanje dokumenata u imenovane kolekcije, po nekom svojstvu
      - slično tabelama u relacionim BP
    - moguća hijerarhija kolekcija

# XML baze podataka

---

- **XML SUBP**
  - upitni jezici
    - **XPath**
      - sa proširenjima za upravljanje kolekcijama dokumenata
    - **XQuery**
      - W3C standard
      - još uvek nije prihvaćen od strane svih XML SUBP

# XML baze podataka

---

- **XML SUBP**
  - **ažuriranje sadržaja**
    - može se izvršiti na više načina
      - ažuriranje celokupnog sadržaja dokumenta
      - ažuriranje dela sadržaja dokumenta
  - jezici za ažuriranje sadržaja
    - XUpdate
      - koristi XPath za odabir elemenata
      - podržava operacije dodavanja, modifikacije i brisanja podataka
    - prošireni XQuery



# XML baze podataka

---

- **XML SUBP**
  - **transakcije, zaključavanje i konkurentno izvršavanje**
    - pojam transakcije je isti kao i u drugim SUBP
      - najmanja jedinica obrade podataka
      - skoro svi XML SUBP podržavaju koncept transakcije
    - nivoi zaključavanja
      - **nivo celog dokumenta**
        - » često slučaj u XML bazama podataka
        - » usporava konkurentan rad
      - **nivo čvora (elementa) u dokumentu**
        - » implementacija ove vrste zaključavanja zahteva zaključavanje nadređenih elemenata, od datog do korena
        - » javlja se problem eksplozije zaključavanja sadržaja

# XML baze podataka

---

- **XML SUBP**
  - **programski pristup**
    - *eng. Application Programming Interfaces (APIs)*
    - ne postoji opšti standard
      - svi XML SUBP imaju svoje API-je
      - definicija metoda za pristup, upite i pretragu metapodataka
    - dva API-ja nezavisna od proizvođača XML SUBP
      - XML:DB API
        - » nezavisan od programskog jezika
        - » koristi XPath
      - JSR 225: XQuery API for Java (XQJ)
        - » zasnovan na JDBC-u
        - » koristi XQuery

# XML baze podataka

---

- **XML SUBP**
  - **očuvanje izvorne strukture dokumenta**
    - *eng. round-tripping*
    - očuvava se izvorna struktura dokumenta, upisana u BP, pri kasnijim čitanjima iz BP
      - praktično, dokument ostaje identičan i po strukturi, a ne samo po sadržaju, pod uslovom da nije u međuvremenu bio menjan
      - vrlo bitno kod XML dokumenata koji opisuju realne dokumente

# XML baze podataka

---

- **XML SUBP**
  - **skladištenje podataka van baze podataka**
    - omogućava skladištenje i referenciranje elemenata koji se ne nalaze u samoj XML bazi podataka
      - npr. nalaze se u relacionoj BP
      - moguće ažuriranje udaljenih podataka posredstvom dokumenata koji se nalaze u XML bazi podataka

# XML baze podataka

---

- **XML SUBP**

- **indeksiranje**

- tipovi indeksa

- **indeksi zasnovani na vrednostima**

- » indeksiraju vrednosti elemenata i atributa

- » npr. *pronađi sve elemente i attribute koji imaju vrednost Vladimir*

- **indeksi zasnovani na strukturi**

- » indeksiraju lokacije elemenata i atributa

- » npr. *pronađi sve radnike*

- **indeksi zasnovani na tekstu**

- » indeksiraju tokene (reči) u vrednostima elemenata i atributa

- » npr. *pronađi sve projekte čije ime sadrži reč 'Projekat'*

- **kombinovani indeksi**

# XML baze podataka

---

- **XML SUBP**
  - **skladištenje spoljašnjih entiteta**
    - skladištenje elemenata koji nisu deo dokumenta
      - predstavljaju elemente koji se pozivaju iz dokumenta
        - » npr. poziv web servisa za vremensku prognozu
          - » skladištenje takvog dokumenta nije primereno
            - » dinamički sadržaj dokumenta – menja se često
    - » npr. refernce na druge dokumente koje čine poglavlja skladištenog dokumenta
      - » skladištenje takvog dokumenta može biti pogodno

# XML baze podataka

---

- **XML SUBP**

- **normalizacija**

- cilj: svaki podatak pojavljuje se tačno jednom
      - pri skladištenju celih dokumenata normalizacija može biti primenjena za uštedu prostora
        - » ukoliko se sadržaji dokumenata preklapaju u većoj meri
  - problem: XML podržava elemente sa više vrednosti
    - kod relacionih baza to ne zadovoljava 1NF
    - normalizacija se ne može vršiti na isti način kao u relacionim BP
  - ne postoji formalan pristup normalizaciji
    - pojam normalizacije ne odgovara pojmu normalizacije u relacionom modelu podataka

# XML baze podataka

---

- **XML SUBP**
  - **ograničenje referencijalnog integriteta**
    - obezbeđuje da pokazivači u XML dokumentima referenciraju validne dokumente ili njihove delove
      - zahteva se validnost i internih i eksternih pokazivača
    - pokazivači mogu biti u obliku
      - ID/IDREF atributi
      - key/keyref elementi
      - XLink-ovi



# XML baze podataka

---

- **Arhitektura XML baze podataka**
  - tipovi arhitektua
    - XML baze podataka zasnovane na tekstu
    - XML baze podataka zasnovane na modelima

# XML baze podataka

---

- **Arhitektura XML baze podataka**
  - **XML baze podataka zasnovane na tekstu**
    - skladišti XML u obliku teksta
    - koristi indekse zasnovane na tekstu
      - za brz pristup i izvršavanje upita
      - jedan pristup disku za indeks + jedan pristup za dokument
    - sporije vraća hijerarhije elemenata

# XML baze podataka

---

- **Arhitektura XML baze podataka**
  - **XML baze podataka zasnovane na modelu**
    - formira i skladišti interni model XML dokumenta
      - npr. DOM model
    - sporije vraćanje celog dokumenta
      - dokument mora da se sastavi od delova, iz internog modela
    - potencijalno brže vraćanje hijerarhije elemenata

# Sadržaj

---

- Skladištenje XML dokumenata u tekstualnom obliku
- Skladištenje XML dokumenata u specijalizovanim XML BP
- Skladištenje sadržaja XML elemenata u specijalizovanim XML BP

# XML baze podataka

---

- **Skladištenje sadržaja XML dokumenata u XML bazi podataka**
  - pogodno ukoliko je dokument polustrukturiran
    - nije moguće izvršiti objektno-relaciono mapiranje

# XML baze podataka

---

- **Skladištenje sadržaja XML dokumenata u XML bazi podataka**
  - prednosti
    - potencijalno veća brzina vraćanja željenih podataka
      - zbog snimanja podataka jednog dokumenta u uzastopne memorijske lokacije
      - nema primene operacije spajanja za vraćanje jednog snimljenog dokumenta
    - mogućnost korišćenja XML upitnih jezika
  - mane
    - manja brzina za vraćanje podataka koji nisu snimljeni kao jedinstven dokument
    - rezultat upita može biti jedino XML dokument

# Sadržaj

---

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- **Upitni jezici**
- XML SUBP

# Upitni jezici

---

- **Tipovi upitnih jezika koji vraćaju rezultat u XML sintaksi**
  - upitni jezik zasnovan na šablonu
  - upitni jezik zasnovan na SQL-u
  - XML upitni jezici



# Upitni jezici

---

- **Upitni jezik zasnovan na šablonu**
  - *eng. template-based query language*
  - većina upita nad relacionom bazom koja vraća XML strukturu zasnovana je na šablonu
  - ne postoji predefinisano mapiranje između XML šeme i šeme baze podataka
    - SQL naredbe su ugrađene u šablon
      - poseban softver izvršava upit i konstruiše rezultujući XML dokument

# Upitni jezici

- **Upitni jezik zasnovan na šablonu**

```
<?xml version="1.0" encoding="UTF-8"?>
<InformacijeLetovi>
  <Uvod>Sledeći letovi su dostupni:</Uvod>
  <SelectStmt>SELECT Prevoznik, BrLeta, Polazak, Dolazak
    FROM Letovi</SelectStmt>
  <Let>
    <Prevoznik>$Prevoznik</Prevoznik>
    <BrLeta>$BrLeta</BrLeta>
    <Polazak>$Polazak</Polazak>
    <Dolazak>$Dolazak</Dolazak>
  </Let>
  <Zakljucak>Srećan put.</Zakljucak>
</InformacijeLetovi>
```

# Upitni jezici

- **Upitni jezik zasnovan na šablonu**

```
<?xml version="1.0" encoding="UTF-8"?>
<InformacijeLetovi>
  <Uvod>Sledeći letovi su dostupni:</Uvod>
  <Let>
    <Prevoznik>ABC Airways</Prevoznik>
    <BrLeta>S425</BrLeta>
    <Polazak>11:15</Polazak>
    <Dolazak>12:00</Dolazak>
  </Let>
  ...
  <Zakljucak>Srećan put.</Zakljucak>
</InformacijeLetovi>
```

# Upitni jezici

---

- **Upitni jezik zasnovan na šablonu**
  - fleksibilan
    - postavljanje vraćenih vrednosti na bilo koje mesto u dokumentu
      - uključujući parametrizovanje narednih SELECT naredbi u istom šablonu
    - mogućnost korišćenja petlji i uslovnih izraza
    - definisanje varijabli i funkcija
    - parametrizovanje SELECT upita HTTP parametrima
  - koriste se isključivo pri transferu podataka iz relacione baze u XML dokument
    - za suprotan smer se koristi mapiranje zasnovano na tabelama

# Upitni jezici

---

- **Upitni jezik zasnovan na SQL-u**
  - koristi modifikovane SQL naredbe
    - čiji rezultat ima XML sintaksu
  - SQL/XML standard
    - 2003. godine
    - uvodi XML tip podataka
    - uvodi XML funkcije

# Upitni jezici

---

- **Upitni jezik zasnovan na SQL-u**

```
SELECT XMLELEMENT(NAME "Narudzbenica",  
                XMLATTRIBUTES(Narudzbenica.NSifra AS NSifra),  
                XMLELEMENT(NAME „Datum", Narudzbenica.Datum),  
                XMLELEMENT(NAME „Klijent", Narudzbenica.Klijent))  
        AS xmldocument  
FROM Narudzbenica
```

```
<Narudzbenica NSifra="123">  
  <Datum>10.08.2012. </Datum>  
  <Klijent> FTN </Klijent>  
</Narudzbenica>
```

# Upitni jezici

---

- **XML upitni jezici**

- mogu da se koriste nad bilo kojim XML dokumentom
  - ne samo nad podacima u bazi podataka
- mogu da se koriste i nad relacionim bazama podataka
  - samo ukoliko su dokumenti modelovani kao XML
    - upiti nad virtuelnim XML dokumentima
- XML upitni jezici
  - XPath
  - XQuery

# Upitni jezici

---

- **XPath**

- namenjen za selekciju jednog ili više **čvorova** u XML dokumentu
- vraćeni elementi moraju da zadovoljavaju definisani **Xpath izraz**
  - zasnovan na konceptu navigacije kroz stablo dokumenta



# Upitni jezici

---

- **XPath**

- osnovni deo XPath izraza je **čvor**
- čvor može da bude
  - element
  - atribut
  - tekst
  - imenski prostor
  - procesna instrukcija
  - komentar
  - dokument čvor

# Upitni jezici

---

- **XPath**

- navigacija kroz stablo dokumenta

- dokument čvor

- početni čvor svakog dokumenta

- korenski element dokumenta

- dete dokument čvora

- roditeljski čvor

- čvor koji se nalazi na prvom višem hijerarhijskom nivou u odnosu na drugi čvor

- » direktno nadređen tom čvoru

- roditeljski čvor može da ima nula, jedno ili više dece

# Upitni jezici

---

- **XPath**

- navigacija kroz stablo dokumenta

- dete čvor

- čvor koji se nalazi na prvom nižem hijerarhijskom nivou u odnosu na drugi čvor

- » direktno podređen tom čvoru

- dete čvor ima tačno jednog roditelja

- čvorovi rođaci

- čvorovi koji imaju istog roditelja

- čvor predak

- bilo koji čvor na višem hijerarhijskom nivou

- » do koga vodi put preko grana stabla

- čvor potomak

- bilo koji čvor na nižem hijerarhijskom nivou

- » do koga vodi put preko grana stabla

# Upitni jezici

---

- **XPath**

- trenutno obrađivani čvor naziva se **tekući čvor**
  - u odnosu na njega se kreće stablom
  - dokument čvor je inicijalni tekući čvor

# Upitni jezici

---

- **XPath**

- separator „/“ odvaja korake u navigaciji po stablu
  - formirajući XPath putanju
  - primer: /korak/korak
- korak putanje sastoji se od
  - ose kretanja
    - u odnosu na tekući čvor
  - naziva čvora
  - nula ili više predikata
    - ograničenja vrednosti traženog čvora
- sintaksa koraka
  - osa::naziv[predikat]

# Upitni jezici

---

- **XPath**

- osa kretanja predstavlja se punim ili skraćenim nazivom

- self:: ili .
    - parent:: ili ..
    - descendant:: ili //
    - attribute ili @
    - child:: podrazumeva se, ako se ništa ne napiše

# Upitni jezici

---

- **XPath**

- zamenski karakteri
  - *eng. wildcards*
  - \* - zamenjuje bilo koji element
  - @\* - zamenjuje bilo koji atribut
  - node() – zamenjuje bilo koji čvor bilo kog tipa
- poseduje operatore koji se mogu koristiti u XPath izrazima
- poseduje ugrađene funkcije

# Upitni jezici

Operator	Opis	Primer	Povratna vrednost
	Unija dva skupa čvorova	//projekat   //radnik	Skup čvorova sa svim projektima i radnicima
+	Sabiranje	6 + 4	10
-	Oduzimanje	6 - 4	2
*	Množenje	6 * 4	24
div	Deljenje	8 div 4	2
mod	Ostatak pri deljenju	5 mod 2	1
=	Jednako	cena=9.80	true ukoliko je cena = 9.80 inače false
!=	Nejednako	cena!=9.80	true ukoliko je cena != 9.80 inače false
<	Manje	cena<9.80	true ukoliko je cena < 9.80 inače false
<=	Manje ili jednako	cena<=9.80	true ukoliko je cena <= 9.80 inače false
>	Veće	cena>9.80	true ukoliko je cena > 9.80 inače false
>=	Veće ili jednako	cena>=9.80	true ukoliko je cena >= 9.80 inače false
or	Disjunkcija	cena=9.80 or cena=9.70	true ukoliko je cena = 9.80 ili cena=9.70, inače false
and	Konjukcija	cena>9.00 and cena<9.90	true ukoliko je cena > 9.00 ili cena<9.90, inače false



# Upitni jezici

---

- **XPath primeri**

/projekti

/projekti/department

//radnik [plata>70000]/Ime

//projekat/\*

//projekat [@\*]

# Upitni jezici

---

- **XQuery**
  - ekvivalent SQL-u
  - koristi XPath za selekciju dela strukture nad kojom se realizuje operacija
  - standardni upitni jezik za XML
    - dizajniran za upite nad XML podacima
      - ne samo nad XML dokumentima
      - može se koristiti nad bazama podataka koje podatke čuvaju u XML obliku

# Upitni jezici

---

- **XQuery**

- tipičan oblik XQuery upita naziva se **FLWR** izraz

FOR <povezivanje varijabli sa pojedinačnim čvorovima>

LET <povezivanje varijabli sa vrednostima>

WHERE <predikat>

RETURN <specifikacija rezultata upita>

# Upitni jezici

---

- **XQuery**
  - klauzule FLWR upita
    - FOR klauzula
      - može se pojaviti nula ili više puta
      - dodeljuje varijabli jedan po jedan element iz sekvence
        - » sekvenca je povratna vrednost izraza putanje
    - LET klauzula
      - može se pojaviti nula ili više puta
      - dodeljuje varijabli jednu vrednost do kraja izvršenja upita
    - WHERE klauzula
      - može se pojaviti nula ili jedan put
      - definiše dodatni uslov nad selekcijom elemenata
    - RETURN klauzula
      - mora se pojaviti tačno jednom
      - definiše izgled vraćenog XML dokumenta

# Upitni jezici

---

- **XQuery**
  - varijable definisane u FOR i LET klauzuli mogu da se koriste u ostatku izraza
    - varijable imaju oblik *\$ime\_varijable*

# Upitni jezici

---

- **XQuery primer**

*Prikazati sve radnike koji rade na projektu sa šifrom 5 više od 20 radnih časova.*

```
LET $d := doc("projekti.xml")
FOR $x IN $d/projekti/projekat[projekatSifra=5]/projekatRadProj,
    $y IN $d/projekti/radnik
WHERE $x/brc gt 20.0 AND $y.JMBG = $x.JMBG
RETURN <res> $y/radnikIme/ime, $y/radnikIme/prezime,
    $x/brc </res>
```

# Upitni jezici

---

- **XQuery primer**

*Prikazati sve radnike koji zarađuju više od 70000 dinara.*

```
FOR $x IN
  doc("projekti.xml")
  //radnik[radnikPlata gt 70000]/radnikIme
RETURN <res> $x/ime, $x/prezime </res>
```

# Upitni jezici

---

- **XQuery primer**

*Prikazati sve radnike koji zarađuju više od 70000 dinara.*

```
FOR $x IN doc("projekti.xml")/projekti/radnik  
WHERE $x/radnikPlata gt 70000  
RETURN <res> $x/radnikIme/ime, $x/radnikIme/prezime </res>
```



# Sadržaj

---

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

# XML SUBP

---

- **eXist**
  - 2000. godina
  - XML SUBP
    - open source
    - kompletno zasnovan na XML tehnologiji
  - podržava XQuery i XPath upitne jezike
  - direktan programski pristup
    - ne zahteva posrednike za pristup iz programskog koda

# XML SUBP

---

- **Sedna**

- 2006. godina
- XML SUBP
  - open source
- podržava XQuery upitni jezike
- pristupa čvorovima XML dokumenta direktno preko pokazivača
  - čuvaju se u internim strukturama
  - veoma brz pristup čvorovima

# XML SUBP

---

- **BaseX**
  - 2007. godina
  - XML SUBP
    - mala veličina
    - procesor XQuery izraza
    - nezavisna od platforme
  - podržava XQuery i XPath upitne jezike
  - podržava vizuelizaciju skladištenih XML dokumenata

# Reference

---

- Elmasri R, Navathe S B, „Fundamentals of Database Systems“, Šesto izdranje, Addison-Wesley, SAD, 2011
  - poglavlje 26
- XML baze podataka
  - <http://www.rpbouret.com/xml/XMLAndDatabases.htm>

# Pitanja i komentari

---



# Sadržaj

- Motivacija
- XML model podataka
- DTD i XML Schema
- XML baze podataka
- Upitni jezici
- XML SUBP

Kraj prezentacije

# Sistemi baza podataka



## XML model podataka

---

*XML Schema jezik i XML baze podataka*