



MHDSVN

Anonymkod/Anonymous code: MHDSVN

# TENTAMEN/EXAMINATION

SDXML VT2024

Modeller och språk för hantering av

Tentamen/Written exam 3,5 hp/hect

IB166N (SU) XMLT AF

Torsdag/Thursday 2024-05-30  
09:00-13:00

Poäng  
Points

Betyg  
Grade

48%

F

Markera besvarade frågor med 'X' / Mark answered questions with 'X'												Antal blad # sheets
1	2	3	4	5	6	7	8	9	10	11	12	
✓	✓	✓	✓	✓	✓							7

Vakt kontrollerat antal blad:

GM.

Obs! Denna sida måste ligga överst - This page should be placed in front  
Avlägsna tomma blad före inlämningen

Remove empty sheets before handing in the exam

Fyll i samtliga uppgifter på sidhuvudet på varje blad

Please fill in all information in the header on each sheet







Stockholms  
universitet

Tentamen/Examination

SDXML

Uppgift nr/Question number

01

Datum/Date

2024/05/30

Blad nr/Page number

01

Anonymkod /  
Anonymous code

MHDSVN

1) Well-formed XML confirms the XML is properly structured according to the rules of extensible markup language. It checks whether rules such as every element is comes in side root Node, elements have proper starting and end elements. Close tags are in the correct order. Those are some of the validations happens. ~~in use~~ If those ~~are~~ requirement are met, It is a well-formed XML document.

2)

3) Processing instruction comes before the root element, It usually Instruct the XML engine about how to process the XML document. ~~The~~ shape of the processing instruction is as bellow.

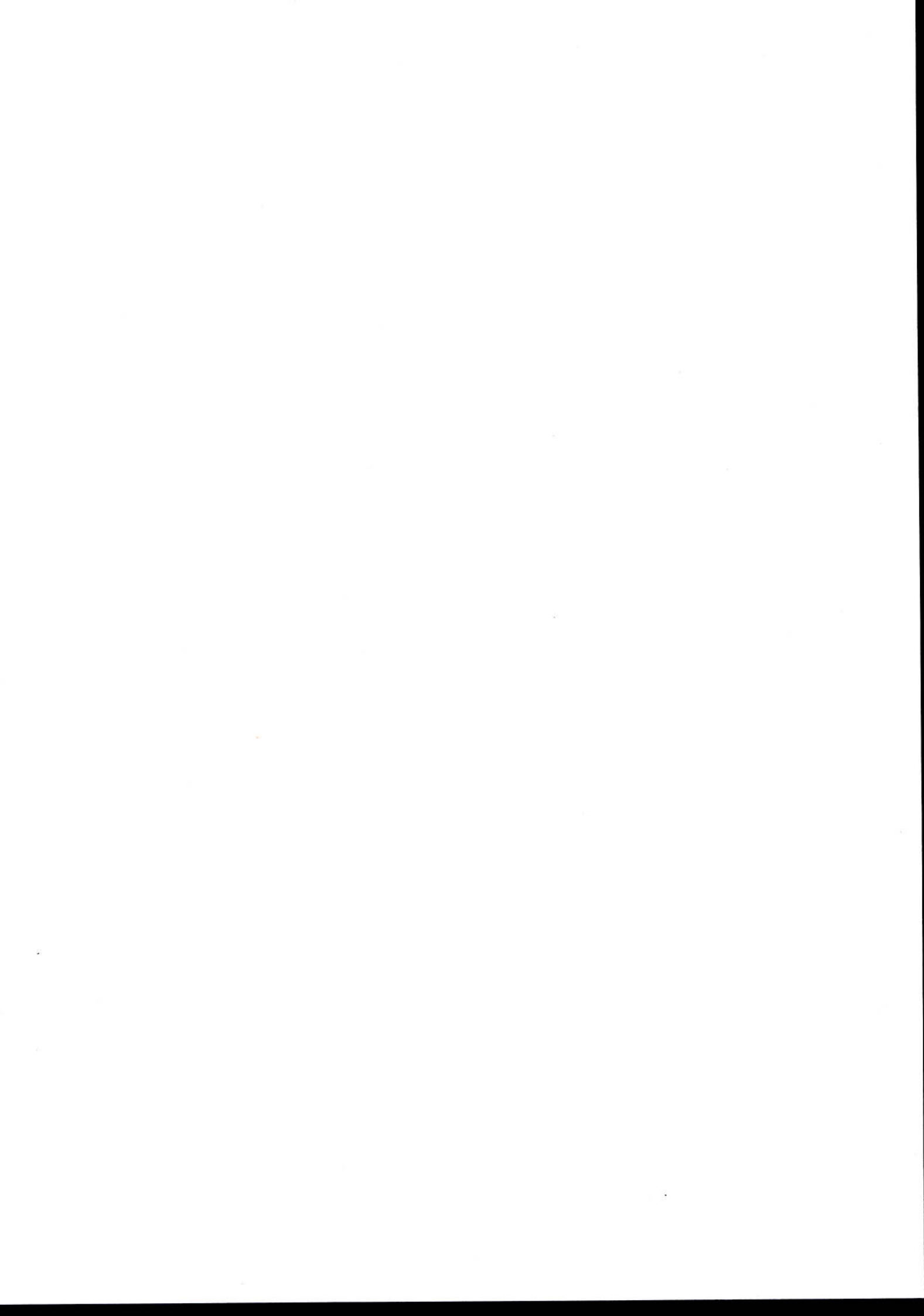
<?name ?>

4) JSON schema itself is a json object which specifies the shape of the json objects. json schema specifies the Attributes of the json, Properties of json object. Required fields, Length of the properties, maximum and minimum values. Whether it accept the additional properties or not.

JSON schema can not be attached to JSON document. Instead the validation can be performed in application level.

5) Sharding is a process of extracting XML document into a atomic values. This is useful when we update XML field of databases. by sharding we can extract some part of the XML to atomic values and update it to the database columns.







Stockholms  
universitet

Tentamen/Examination

SDXML

Datum/Date

2024/05/30

Uppgift nr/Question number

01

Blad nr/Page number

02

Anonymkod /  
Anonymous code

MHDSVN

- 6) Scalr Vector graphic is an XML based Language which is used for create Vector graphic such as basic shapes such as circles, squares, or complex shapes using coordinates. It allows us to color shapes. It will simply allows us to create graphics using XML like language.







Stockholms  
universitet

Tentamen/Examination

SDXML

Datum/Date

2024/05/30

Uppgift nr/Question number

02

Blad nr/Page number

03

Anonymkod /

Anonymous code

MHDSVN

2-  
{  
 "name": "pedik",  
 "colors": ["white", "brown", "yellow"],  
 "sizes": [  
 {  
 "code": "0001",  
 "price": 220  
 },  
 {  
 "code": "0002",  
 "price": 110,  
 "available": true  
 }  
 ],  
 "available": true  
}







Stockholms  
universitet

Tentamen/Examination

SDXML

Uppgift nr/Question number

03

Datum/Date

2024/05/30

Blad nr/Page number

04

Anonymkod /  
Anonymous code

MHDSVN

< Parks >

< Park name = "Park 1" size = "1000" >

redundancy

< TreeType name = "treeType1" maximumHeight = "10" averageLife  
inYears = "20" >

< TreeTypeInPark number = "1" planted = "2000-01-02" / >

< TreeTypeInPark number = "2" planted = "2002-03-10" / >

< TreeTypeInPark number = "3" planted = "1989-12-30" / >

< /TreeType >

< TreeType name = "Orke" maximumHeight = "20" averageLifeIn  
Years = "70" >

< TreeTypeInPark number = "1" planted = "1994-03-09" / >

< TreeTypeInPark number = "2" planted = "1996-04-12" / >

< /TreeType >

\* < TrashCan volume = "2" longitude = "001415" latitude = "220001" / >

< TrashCan volume = "5" longitude = "2001459" latitude = "321087" / >

< /Park >

< Park name = "Park 2" size = "10500" >

< TreeType name = "Willow" maximumHeight = "30" averageLifeIn  
years = "30" >

< TreeTypeInPark number = "10" planted = "1993-12-30" / >

< TreeTypeInPark number = "11" planted = "1990-12-25" / >

< /TreeType >

< /Park >

< /Parks >

Tree types  
in no parks?

Schema?

L-







Stockholms  
universitet

Tentamen/Examination

SDXML

Uppgift nr/Question number

04

Datum/Date

2024/05/30

Blad nr/Page number

05

Anonymkod /  
Anonymous code

MHDSVN

2

```
(a) element Genres {  
  for $gn in distinct-values (//Genre)  
  let $ml:=for $m in //Movie[Genre=$gn]  
    return element Movie {  
      attribute Title {data$m/@Title}  
      attribute NumberOfOtherGenres {  
        count($m/Genre[text() = not(text()=$gn)]  
      }  
    }  
  }  
  return element Genre {  
    attribute Name { $gn },  
    $ml  
  }  
}
```

2

```
(b) element Result {  
  for $gn in distinct-values (//Genre)  
  let $shos:=for $sh in //Movie[Genre=$gn]//showing  
    return element Showing {  
      attribute star $sh/@starttime,  
      attribute movie { $sh/../../@title },  
      attribute cinema { $sh/../../@name },  
      $sh/@hall  
    }  
  return element Genre {  
    attribute name { $gn },  
    $shos  
  }  
}
```







Stockholms  
universitet

Tentamen/Examination

SDXML

Uppgift nr/Question number

~~04~~ 05

Datum/Date

2024/05/30

Blad nr/Page number

06

Anonymkod /  
Anonymous code

MHDSVN

(3) 

```
<xsl:transform version="1.0" xmlns="" >
  <xsl:output method="html"/>
  <xsl:template match="/">
    <xsl:apply-templates select="//Cinema[not(
      @name=pr following::Cinema/@name)]>
    </xsl:apply-templates>
  <xsl:template>
    <xsl:template match="Cinema">
```

0







Stockholms  
universitet

Tentamen/Examination

SDXML

Uppgift nr/Question number

06

Datum/Date

2024/05/30

Blad nr/Page number

07

Anonymkod /  
Anonymous code

MHDSVN

(a) ~~select~~ ~~code~~ Code as "Code",  
at.name as "Name",  
XMLQuery ("~~max~~max(\$r//Rate)" passing ~~as~~ rates as 'r')  
as "HighestRateEver"

from Bank b inner join AccountType at inner join b.code  
= at.bank

How many rows per bank?

0 ~~not~~

(b) select XMLElement(NAME "Result",  
~~xmlagg~~ ~~xmlagg~~ xMLElement(NAME "Bank",  
xmlattributes(name as "Name", code as  
"Code"),  
XMLElement(NAME "AccountType",  
~~xmlattributes~~  
xmlattributes(at.name as "Name"))  
)

XMLAGG?

from Bank b inner join AccountType at on  
b.code = at.bank

freeWithdrawals?

0,5

