

程序代写代做 CS编程辅导

---



# COMP3349 Web Data Compression and Search

WeChat: cstutorcs  
Assignment Project Exam Help

Semistructured / Tree Data,

QQ: 749389476  
Email: tutorcs@163.com

<https://tutorcs.com>

# Semistructured Data 程序代写代做CS编程辅导

■ Emails, HTML, JSON, XML, RDF, ...



Unstructured text

WeChat: cstutorcs

Structured, Unstructured and Semi-Structured

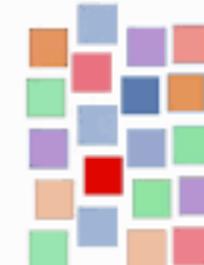
Semi-Structured Data



Structured Data



Unstructured Data

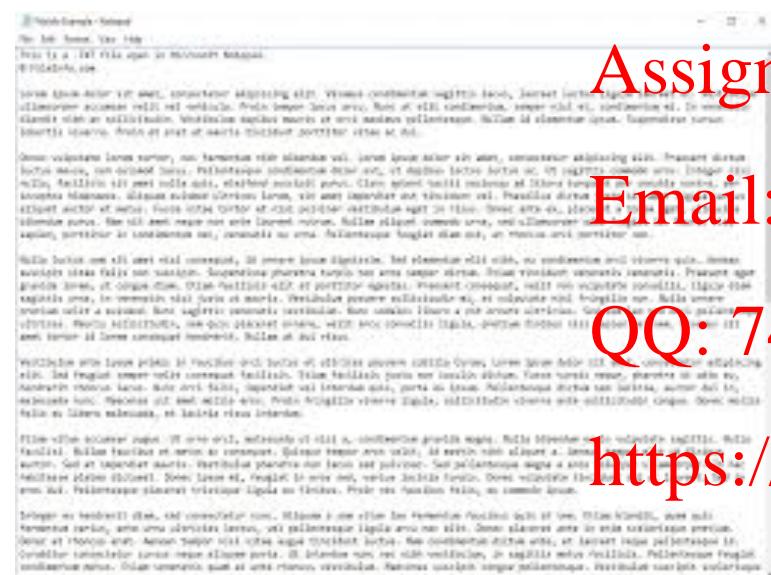


Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



# JSON

程序代写代做 CS编程辅导



```
{  
    "orders": [  
        {  
            "orderno": "748745375",  
            "date": "June 30, 2088 1:54:23 AM",  
            "trackingno": "TN0039291",  
            "custid": "11045",  
            "customer": [  
                {  
                    "custid": "11045",  
                    "fname": "Sue",  
                    "lname": "Hartfield",  
                    "address": "1109 Silver Street",  
                    "city": "Ashland",  
                    "state": "ME",  
                    "zip": "68003"  
                }  
            ]  
        }  
    ]  
}
```

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# HTML

程序代写代做 CS编程辅导



html5temp.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <title> A Tiny HTML Document </title>
6  <link href = "style.css" rel="stylesheet">
7  <script src="scripts.js"></script>
8  </head>
9
10 <body>
11 <p>Let's rock the browser - HTML5 style.</p>
12 </body>
13 </html>
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# RDF

程序代写代做 CS编程辅导



```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/documents/1998/09/dces/#">
    <rdf:Description>
        <dc:title>Flag of Algeria</dc:title>
        <dc:creator>BISHI Computer</dc:creator>
        <dc:subject>Country Flags</dc:subject>
        <dc:date>October 2001</dc:date>
        <dc:color>red, green, white</dc:color>
        <dc:feature>cresent moon, star</dc:feature>
    </rdf:Description>
</rdf:RDF>
```

WeChat: cstutorcs  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# XML

程序代写代做 CS编程辅导



```
▼<root>
  ▼<Pro
    ▼<Product>
      <Code>2941</Code>
      <StockQty>65</StockQty>
      <Barcode>49020570284087</Barcode>
    </Product>
    ▼<Product>
      <Code>2778</Code>
      <StockQty>200</StockQty>
      <Barcode>490205700443053</Barcode>
    </Product>
    ▼<Product>
      <Code>2838</Code>
      <StockQty>140</StockQty>
      <Barcode>4902057003726</Barcode>
    </Product>
  </Products>
</root>
```

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Semistructured Data / JSON / XML / ...

程序代写代做 CS编程辅导



## ■ Semistructured Data

- | loosely structured (no restrictions on tags & nesting relationships)

WeChat: cstutorcs

- | no schema required

Assignment Project Exam Help

## ■ XML / JSON / ...

Email: tutorcs@163.com

- | under the “semistructured” umbrella

QQ: 749389476

- | self-describing

<https://tutorcs.com>

- | the standard for information representation & exchange

# Web Data in COMP310



- We assume XML form, since:
  - HTML, RDF, XML, ... ∈ XML
  - Other semistructured data such as JSON, Emails, ... can be easily mapped to XML

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做CS编程辅导

**XML**



**XML (*eXtensible Markup Language*) is a standard developed by W3C (World Wide Web Consortium) and endorsed by a host of industry heavyweights such as IBM, Microsoft, SAP, Software AG, General Motors, ...**

<https://tutorcs.com>

# Storage format vs presentation format - The power of markup



Raymond, \tatabase or Spreadsheet

x5932, John, Smith, jsmith, 1234, ...

## HTML

```
<br>
<font size=1 color="ff003a">
<ul>
<li> <b> Raymond Wong </b> </li>
<li> Login: wong </li>
<li> Phone: <i> x5932 </i> </li>
</ul>
</font>
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

## XML

```
<Staff>
  <Name>
    <FirstName> Raymond </FirstName>
    <LastName> Wong </LastName>
  </Name>
  <Login> wong </Login>
  <Ext> 5932 </Ext>
</Staff>
```

# XML Terminology



- tags: book, title, author, ...
- start tag: <book>, end tag: </book>
- elements: <book>...</book>, <author>...</author>  
Assignment Project Exam Help
- elements are nested
- empty element: <red></red> abbr. <red/>
- an XML document: ~~QQ: 749389476~~ root element
- well formed XML document: if it has matching tags  
<https://tutorcs.com>

# Resources

程序代写代做 CS 编程辅导



- [www.w3.org](http://www.w3.org)
- [www.xml.com](http://www.xml.com)
- [www.xml.org](http://www.xml.org) WeChat: cstutorcs
- [www.oasis-open.org](http://www.oasis-open.org) Assignment Project Exam Help
- [Email: tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476

<https://tutorcs.com>

# More XML: Attributes



```
<book price = "55" currency = "USD">  
    <title> Foundations of Databases </title>  
    <author> Abiteboul </author>  
    ...  
    Email: tutorcs@163.com  
    <year> 1995 </year>  
    QQ: 749389476  
</book> https://tutorcs.com
```

# More XML: Oids and References



```
<person id="o123">
    <name>John</name>
    </person>
    WeChat: cstutorcs
<person id="o456">
    <name>Mary</name>
    Assignment Project Exam Help
    <child>Tom</child>
    Email: tutorcs@163.com
    </person>
    QQ: 749389476
<person id="o123" mother="o456">
    https://tutorcs.com
    <name>John</name>
    </person>
```

# XML/JSON/semistructured data can be modeled in a tree form

```
<Staff>
  <Name>
    <FirstName> Raymond
    <LastName> Wong </LastName>
  </Name>
  <Login> wong </Login>
  <Ext> 5932 </Ext>
</Staff>
```



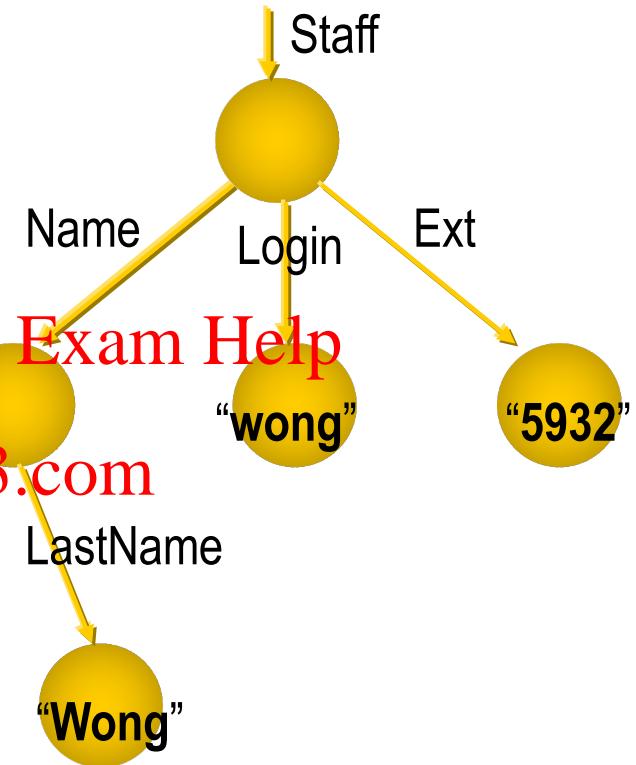
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



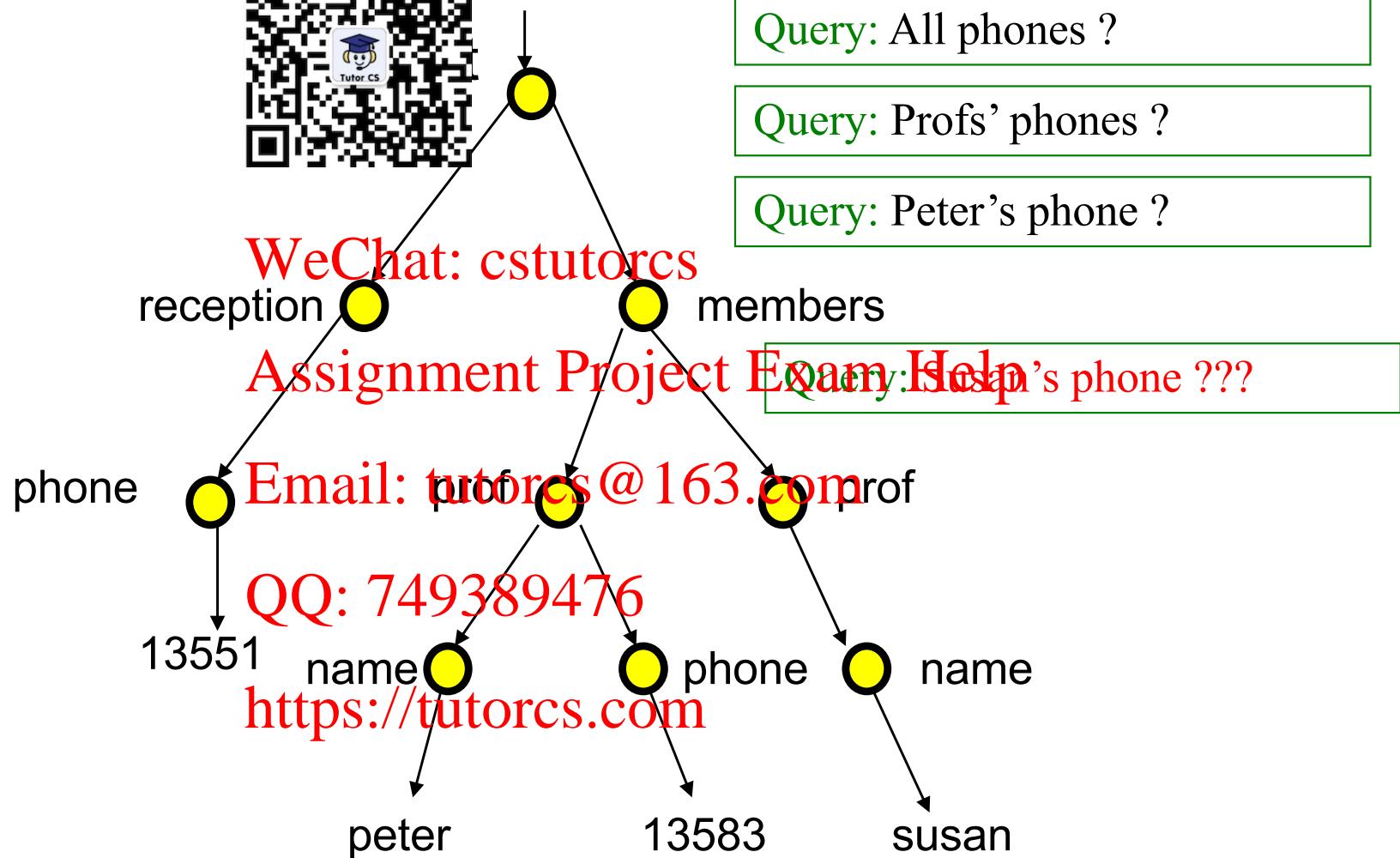
# Why need to query tree data



- To extract data from a large tree
- To exchange data (data- or query-shipping)  
WeChat: cstutorcs
- To exchange data between different user  
Assignment Project Exam Help  
communities or ontologies or schemas  
Email: [tutorcs@163.com](mailto:tutorcs@163.com)
- To integrate data from multiple data sources  
QQ: 749389476

<https://tutorcs.com>

# Answering queries requiring navigation of the data tree



# XPath 1.0

程序代写代做 CS编程辅导



- <http://www.tutorcs.com/TR/xpath> (11/99)
- Building block for other W3C standards:
  - XSL Transformations (XSLT)
  - XML Link (XLink)
  - XML Pointer (XPointer)
  - XPath 2.0   QQ: 749389476
  - XQuery        <https://tutorcs.com>
- Was originally part of XSL

# Example for XPath Queries



```
<bib>
  <book>    <publisher> Addison Wesley </publisher>
             <author> Steven T. Hull </author>
             <author> <first-name> Rick </first-name>
                       <last-name> Hull </last-name>
             </author> WeChat: cstutorcs
             <author> Victor Vianu </author>
             <title> Foundations of Databases </title>
             <year> 1995 </year>
  </book>          Assignment Project Exam Help
  <book price="55">
    <publisher> Freeman </publisher>
    <author> Jeffrey D. Ullman </author>
    <title> Principles of Database and Knowledge Base Systems </title>
    <year> 1998 </year>
  </book>
</bib>
```

Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476  
<https://tutorcs.com>

# Data Model for XPath



WeChat: cstutorcs The root element

Assignment Project Exam Help

book book

Email: tutorcs@163.com

QQ: 749389476

author https://tutorcs.com

publisher

Addison-Wesley

Serge Abiteboul

# XPath: Simple Expressions

/bib/book/year



Result: <year> 1995 </year>  
<year> 1998 </year>

Email: tutorcs@163.com

/bib/paper/year QQ: 749389476

<https://tutorcs.com>

Result: empty

# XPath: Restricted Kleene Closure

程序代写代做 CS编程辅导

//author



Result: <author> biteboul </author>  
<author> <first-name> Rick </first-name>  
WeChat: cstutorcs <last-name> Hull </last-name>  
</author> Assignment Project Exam Help  
<author> Victor Vianu </author>  
Email: tutorcs@163.com  
<author> Jeffrey D. Ullman </author>  
QQ: 749389476

/bib//first-name <https://tutorcs.com>

Result: <first-name> Rick </first-name>

# XPath: Text Nodes



/bib/book/author

Result: Serge Abiteboul

Victor Vian WeChat: cstutorcs

Jeffrey D. Ullman

Rick Hull doesn't appear because he has *firstname, lastname*

Assignment Project Exam Help  
Functions in XPath: Email: tutorcs@163.com

- | text() = matches the text value  
QQ: 749389476
- | node() = matches any node (= \* or @\* or text())
- | name() = returns the name of the current tag

# XPath: Wildcard

//author/\*



WeChat: cstutorcs

Result: <first-name> Rick </first-name>  
<last-name> Hull </last-name>  
Assignment Project Exam Help  
Email: tutorcs@163.com

QQ: 749389476

\* Matches any element  
<https://tutorcs.com>

# XPath: Attribute Nodes



/bib/book/@p

Result: “55”

@price means that price is has to be an  
attribute

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# XPath: Qualifiers

/bib/book/author[1]/name]



Result: <author> <first-name> Rick </first-name>  
<last-name> Hu </last-name>  
</author>  
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# XPath: More Qualifiers



/bib/book/author[first-name='John']/address[//zip][city]]/lastname

Result: <lastname> ... <WeChat>tutorcs

<lastname> ... </lastname> Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# XPath: More Qualifiers



/bib/book[@pi  
“60”]

WeChat: cstutorcs  
/bib/book[author/@age < “25”]  
Assignment Project Exam Help

Email: tutorcs@163.com  
/bib/book[author/text()]  
QQ: 749389476

<https://tutorcs.com>

# XPath: More Details



We can navigate along the tree using axes:

ancestor

ancestor-or-self

attribute

child

descendant

descendant-or-self

following

following-sibling

namespace

parent

preceding

preceding-sibling

self

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Differences from traditional DB



- What sets semi-structured/XML data servers apart from RDBMS or OODB is the lack of typing.  
WeChat: cstutorcs
- This affects mostly the way the data is stored and indexed.  
Email: tutorcs@163.com
- Also, Web data are inherently distributed  
QQ: 749389476

<https://tutorcs.com>

# Implementing XML Repository



## Repository based on

- | plain text file
- | relational database
- | object database
- | tailor-made, specialized XML database

## Type information

- | even partial typing information can be used to improve the storage

# Text files

程序代写代做 CS编程辅导

- *it's the simplest way to store*
- *easy to handle*
- *widely available*
- have to check out an entire doc in order  
to retrieve a datum  
WeChat: cstutorcs  
Assignment Project Exam Help
- simultaneously access/update
- access/modify an item from a large  
catalog collection  
Email: tutorcs@163.com  
QQ: 749389476  
<https://tutorcs.com>



# Relational databases



- existing, proven technology to provide full database management

WeChat: cstutorcs

Assignment Project Exam Help

- it's not easy and efficient to manage XML data in traditional RDBMS

<https://tutorcs.com>

# An Example (using RDBMS)



- assume no tree information
- data can be an arbitrary graph  
WeChat: cstutorcs
- let's use two tables for the XML instances:
  - one to store all edge information
  - one to store values

Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476

<https://tutorcs.com>

# The two tables



Ref(src, label,

Val(oid, value)

WeChat: cstutorcs

Assignment Project Exam Help

Suppose a simple query like:

Email: tutorcs@163.com

family/person/hobby

QQ: 749389476

in XPath

<https://tutorcs.com>

# The same query in SQL

```
select v.value
```



```
from Ref r1, Re
```

```
where r1.src = “root” AND r1.label = “family”
```

```
AND r1.dst = r2.src AND r2.label = “person”
```

```
AND r2.dst = r3.src AND r3.label = “hobby”
```

```
AND r3.dst = v.oid
```

Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476

This is a 4-way join!!!  
<https://tutorcs.com>

It's very inefficient though index on label can help a lot.

# Efficiency problem 程序代写代做CS编程辅导



- even simple will have a large no of joins
- RDBMS organizes data based on the structure of tables and type info => clustering, indexing, query optimization are not working properly for XML data
  - WeChat: cstutorcs
  - Assignment Project Exam Help
  - Email: tutorcs@163.com
  - QQ: 749389476
- Also #ways to traverse path expressions are much more than that on tables
  - <https://tutorcs.com>