

CS157A:
Introduction to Database
Management Systems
Assignment Project Exam Help

<https://powcoder.com>

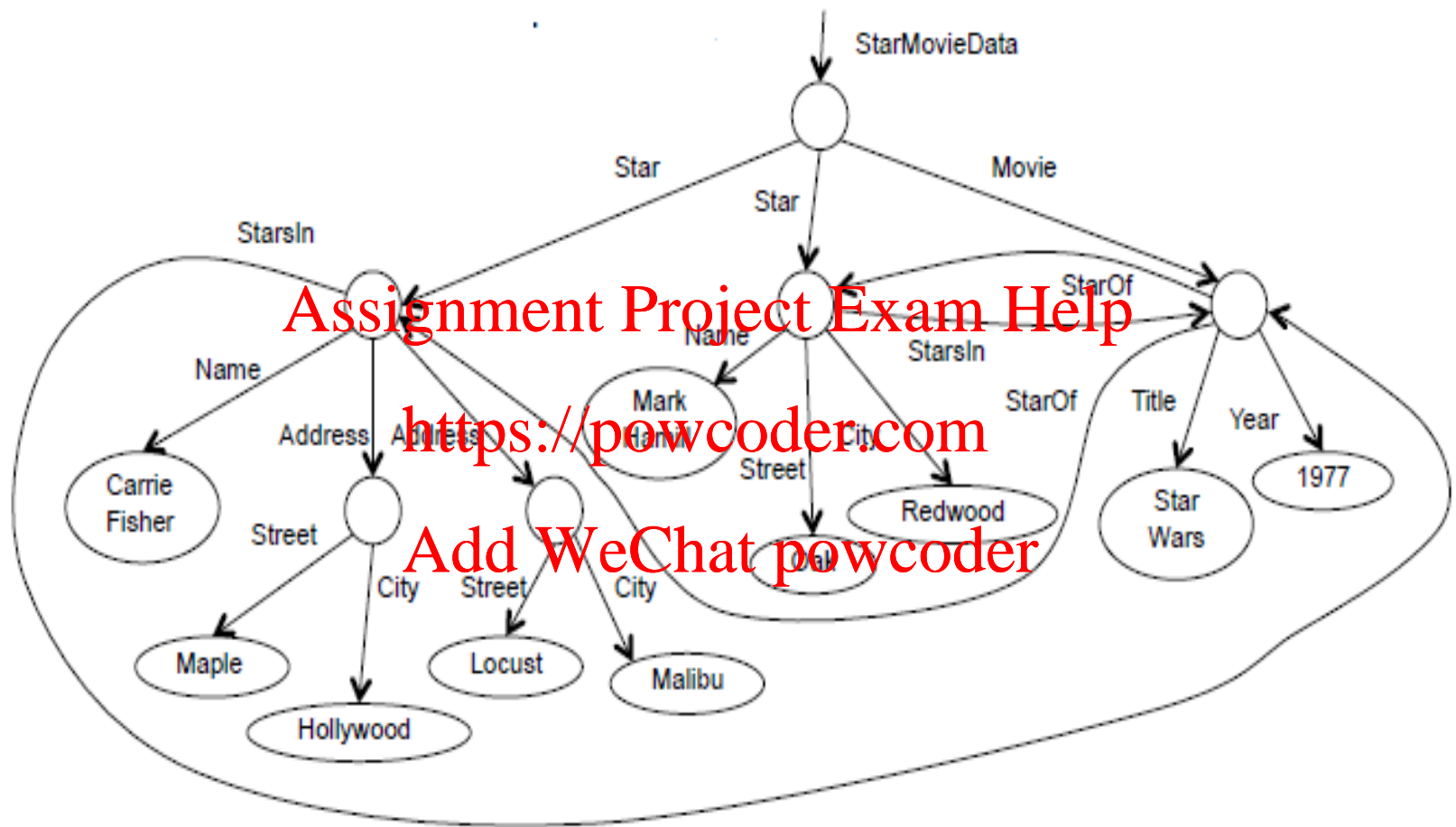
Add WeChat powcoder

Chapter 11: XML

Suneuy Kim

Semi structured data representation

- A database of semi structured data is a hierarchical collection of nodes.
- Root represents the entire database.
- Immediate children of roots represents central entities.
- Leaf nodes have data
- A label on an arc from node N to node M
 - name of the attribute or the sub element
 - relationship



Semi-Structured Data Model

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<StarMovieData>
```

```
  <Star starID = "cf" starredIn = "sw">
```

```
    <Name>Carrie Fisher</Name>
```

```
    <Address>
```

```
      <Street>123 Maple St.</Street>
```

```
      <City>Holly wood</City>
```

```
    </Address>
```

```
    <Address>
```

```
      <Street>5 Locus Ln.</Street>
```

```
      <City>Malibu</City>
```

```
    </Address>
```

```
  </Star>
```

```
  <Star starID = "mh" starredIn = "sw">
```

```
    <Name>Mark Hamil </Name>
```

```
    <Street>456 Oak Rd.</Street>
```

```
    <City>Brentwood</City>
```

```
  </Star>
```

```
  <Movie movieID="sw" starsOf = "cf mh">
```

```
    <Title>Star Wars</Title>
```

```
    <Year>1977</Year>
```

```
  </Movie>
```

```
</StarMovieData>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

StarMovieData.xml - XML Data
corresponding to the Semi-Structured
Data Model on pp. 3.

XML (Extensible Markup Language)

- Standard for data representation and exchange
- Basic constructs
 - Tagged elements (can be nested)
 - Attributes
 - Text
- Tags
 - Play the same role as the labels on the arcs of semi structured-data graph.
 - HTML tags describe formatting
 - XML tags describe content, that is, meaning of data

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example: XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<Movies>
```

```
  <Movie title = "King Kong">
```

```
    <Version year = "1933">
```

```
      <Star>Far Wray</Star>
```

```
    </Version>
```

```
    <Version year = "1976">
```

```
      <Star>Carrie Fisher</Star>
```

```
      <Star>Jessica Lange</Star>
```

```
    </Version>
```

```
  </Movie>
```

```
  <Movie title = "Footloose">
```

```
    <Version year = "1984">
```

```
      <Star> Kevin Bacon</Star>
```

```
      <Star>John Lithgow</Star>
```

```
      <Star>Sarah Jessica Parker</Star>
```

```
    </Version>
```

```
  </Movie>
```

```
</Movies>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Semantic Tag

- Tags are normally matched pairs, as

```
<Movie> . . . </Movie>
```

- Opening tag can have attributes.

```
<Movie title = "King Kong">
```

- Tags may be nested arbitrarily.

- Element - A pair of matching tags and everything that comes between them.

```
<Version year = "1933">  
  <Star>Far aWay</Star>  
</Version>
```

- A **single tag** is used for an element that doesn't have any sub-element. A single tag can have attributes.

```
<Movie title="Star Wars" year = "1977"/>
```

- XML tags are case-sensitive.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Attributes

- An alternative way to represent a leaf node

```
<Movie year = "1977">
```

```
  <Title>Star Wars </Title>
```

```
</Movie>
```

```
<Movie year = "1977" title = "Star Wars" >
```

```
</Movie>
```

```
<Movie year = "1977" title = "Star Wars" />
```

- Identifier of an element

```
<Star starID = "cf" starredIn = "sw"> </Star>
```

```
<Star starID = "mh" starredIn = "sw" > </Star>
```

- To connect elements

```
<Movie movieID = "sw" starsOf = "cf mh" > </Movie>
```


Namespaces

To distinguish among different vocabularies for tags in the same document

Assignment Project Exam Help

<md:StarMovieData https://powcoder.com

"http://infolab.stanford.edu/movies">

Add WeChat powcoder

URI: URL that refers to a document describing the meaning of the tags in the name space.

XML with and without a Schema

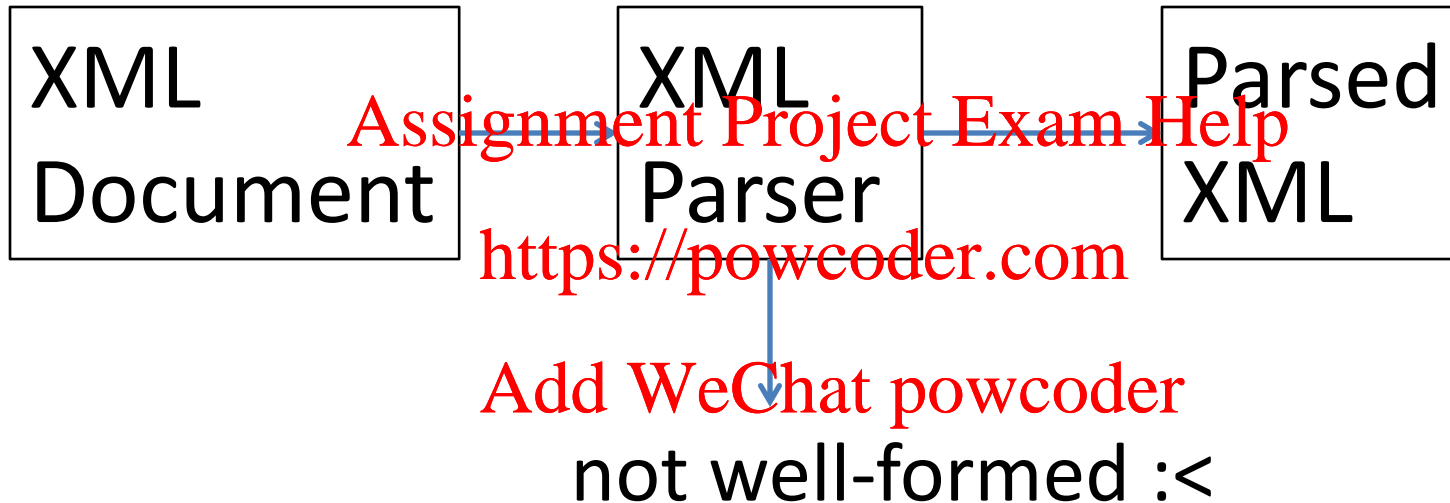
- Well-formed XML
 - You can invent your own tags – no predefined schema
 - The nesting rule for tags must be obeyed.
- Valid XML <https://powcoder.com>
 - Conforms to a certain DTD (Document Type Definition) or a XML Schema
 - DTD/XML Schema specifies the allowable tags and a grammar about how they may be nested.

Well-formed XML

An XML document is called well-formed if it satisfies the following rules, specified by the W3C.

- A well-formed XML document must have a corresponding end tag for all of its start tags.
- Nesting of elements within each other in an XML document must be proper. For example, `<tutorial><topic>XML</topic></tutorial>` is a correct way of nesting but `<tutorial><topic>XML</tutorial></topic>` is not.
- In each element two attributes must not have the same name. For example, `<tutorial id="001"><topic>XML</topic></tutorial>` is right, but `<tutorial id="001" id="w3r"><topic>XML</topic></tutorial>` is incorrect.
- Markup characters must be properly specified.
- An XML document can contain only one root element. So, the root element of an xml document is an element which is present only once in an xml document and it does not appear as a child element within any other element.

Well-Formed XML



Valid XML

- Adheres to basic structural requirements
- Adheres to content-specific specification
 - Document Type Descriptor (DTD)
 - XML Schema (XSD)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Valid XML

DTD/XML Schema

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

XML
Document

Validating
XML
Parser

Parsed
XML

not well-formed or
not valid

Valid vs. Well-formed XML

- Valid XML – benefit of typing
 - Application programs can assume structure
 - DTD/XSD can serve as specification for data exchange
 - Documentation
- Well-formed XML – flexibility, benefit of no-typing
 - Flexibility – ease of change
 - DTD/XSD can be messy for irregular data

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

DTD (Document Type Definitions)

- Language to describe XML schema by specifying elements, attributes, nesting, ordering and # of occurrences
- Also special attribute types for key and foreign key(s): ID and IDREF(s)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

The form of a DTD

```
<!DOCTYPE root-tag [  
  <!ELEMENT element-name(components)>  
  ... more elements...  
>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

DTD Elements

- The description of an element consists of its name (tag), and a parenthesis containing any nested tags.
- Sub tags must appear in order shown
- Each tag may be followed by its multiplicity.
 - A*: any number of times including 0
 - A+: one or more times
 - A?: either zero or one time, but no more
- Symbol | can connect alternative sequences of tags. Example: (A|B) means A or B, but not both.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

DTD Elements: #PCDATA and EMPTY

- Leaves (text elements) have #PCDATA (*Parsed Character DATA*) in place of nested tags
<https://powcoder.com>
[Add WeChat powcoder](#)
– The element has a text value and no nested element within it.
e.g.) `<!ELEMENT Title (#PCDATA)>`
- `<!ELEMENT Foo EMPTY>` means `<Foo />` or `<Foo></Foo>` is the only available form of Foo.

Example: DTD Elements

<!ELEMENT Genre (Comedy| Drama|SciFi|Teen)>
<!ELEMENT Address (Street, (City|Zip))>
<!ELEMENT NAME ((TITLE?, FIRST, LAST) | IPADDR)>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Using a DTD

1. Set standalone = "no".
2. Either:
 - a) Internally include the DTD as a preamble of the XML document, or
 - b) Follow DOCTYPE and the <root tag> by SYSTEM and a path to the external file where the DTD can be found.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example: (a) InternalDTD.xml

```
<?xml version = "1.0" standalone = "no" ?>
```

```
<!DOCTYPE Stars[
```

```
  <!ELEMENT Stars(Star*)>
```

```
  <!ELEMENT Star (Name, Address+)>
```

```
  <!ELEMENT Name (#PCDATA)>
```

```
  <!ELEMENT Address(Street City)>
```

```
  <!ELEMENT Street (#PCDATA)>
```

```
  <!ELEMENT City (#PCDATA)>
```

```
<Stars>
```

```
  <Star><Name>Carrie Fisher</Name>
```

```
    <Address><Street>123 Maple St. </Street> <City>Holly Wood</City></Address>
```

```
    <Address><Street>5 Locust Ln.</Street> <City>Malibu</City></Address>
```

```
  </Star>
```

```
  <Star> ...
```

```
</Stars>
```

← The DTD

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

↗ The XML
document

Example: (b) ExternalDTD.xml

- Assume the Stars DTD is in file default.dtd.

```
<?xml version = "1.0" standalone = "no" ?>
```

```
<!DOCTYPE Stars SYSTEM "default.dtd">
```

```
<Stars> Assignment Project Exam Help
```

```
  <Star><Name>Carrie Fisher</Name>
```

```
    <Address><Street>123 Maple St. </Street>
```

```
      <City>Holly Wood</City></Address>
```

```
    <Address><Street>5 Locust Ln. </Street>
```

```
      <City>Malibu</City></Address>
```

```
  </Star>
```

```
  <Star> ...
```

```
</Stars>
```

Internal vs. External DTD

External DTD are better because of:

- possibility of sharing definitions between XML documents
- The documents that share the same DTD are more uniform and easier to retrieve

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Attributes

- Opening tags in XML can have *attributes*.

- In a DTD, [Assignment Project Exam Help](https://powcoder.com)

`<!ATTLIST E ... >`

<https://powcoder.com>

declares attributes for element *E*, along with its data type.

[Add WeChat powcoder](https://powcoder.com)

Attributes

DTD:

```
<!ELEMENT Movie EMPTY>
```

```
<!ATTLIST Movie
```

```
  title CDATA #REQUIRED
```

```
  year CDATA #REQUIRED
```

```
  genre (comedy | drama | sciFi | teen) #IMPLIED>
```

XML:

```
<Movie title = "Star Wars" year = "1977" genre =  
"sciFi"/>
```

Example: ATTLIST in DTD

- MoviesWithAttribute.dtd
- MoviesWithAttribute.xml

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

DTD types: ID and IDREF

DTD:

```
<!ATTLIST Star  
  starID ID #REQUIRED  
  starredIn IDREF #IMPLIED >
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

XML:

```
<Star starID = "cf" starredIn "sw">
```

DTD types: ID and IDREF

DTD:

```
<!ATTLIST Movie
```

```
  movield ID #REQUIRED
```

```
  starsOf IDREFS #IMPLIED
```

```
>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

XML:

```
<Movie movield = "sw" starsOf = "cf mh">
```

Example: ID and IDREF

- StarMovieData.dtd
- StarMovieData.xml

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Structure of an XML-Schema Document

<? xml version = ... ?>

<xs:schema xmlns:xs =
"http://www.w3.org/2001/XMLSchema">

</xs:schema>

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Defines "xs" to be the namespace described in the URL shown.

Interpret the meaning of schema as part of the name space xs.

Elements of XML Schema

- `<xs:element name = "...\" type = "...\" />`
 - name: the tag-name of the element being defined.
 - type: the type of the element.
 - Simple type e.g., `xs:string`, `xs:integer`, and `xs:boolean`
 - Complex type and Restricted Simple type that are defined in the document itself
- Use `minOccurs` and `maxOccurs` attributes to control the number of occurrences of an `xs:element`.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

minOccurs and maxOccurs

- minOccurs: no fewer than minOccurs
- maxOccurs: no more than maxOccurs
- If there is more than one, they must all appear consecutively.
- Unbounded: no upper bound limit
- Default is one occurrence.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

xs:element

In XML Schema:

```
<xs:element name = "Title" type = "xs:string" />
```

```
<xs:element name = "Year" type = "xs:integer" />
```

<https://powcoder.com>

Add WeChat powcoder

XML Elements:

```
<Title> Star Wars</Title>
```

```
<Year> 1977 </Year>
```

User-defined Types

- Complex Types - to define a complex type using existing types
- Restricted Simple Type - to define a simple type by restricting a base type
 - enumerations
 - range-restricted base types

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Complex Types

Several ways to construct a complex type

- xs:sequence – order matters
- xs:all – the child elements can appear in any order and that all of the child elements occur once or none of them occur.
<https://powcoder.com>
- xs:choice – any one of the elements will appear
[Add WeChat powcoder](#)

Complex Types

name of the complex type

`<xs:complexType name = "movieType">`

`<xs:sequence>`

typical sub-element of complex type

`<xs:element name = "Title" type = "xs:string"/>`

`<xs:element name = "Year" type = "xs:integer"/>`

`</xs:sequence>`

`</xs:complexType>`

Note: you need a name if you want to use it for the type of multiple elements.

Alternative: Complex Types defined in an Element

```
<xs:element name = "Movies">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "Movie" type = "movieType"
        minOccurs = "0" maxOccurs = "unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

no type attribute

type of element Movies,

no type name

<https://powcoder.com>

Add WeChat powcoder

A DTD for Movies

```
<!DOCTYPE Movies [  
  <!ELEMENT Movies (Movie*)>  
  <!ELEMENT Movie (Title, Year)>  
  <!ELEMENT Title (#PCDATA)>  
  <!ELEMENT Year (#PCDATA)>  
>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example

- MoviesValidatedBySchema.xml
 - MoviesValidatedBySchema.xsd
 - MoviesValidatedBySchema.dtd
- <https://powcoder.com>

Add WeChat powcoder

Example: xs:all

```
<xs:element name="person">
  <xs:complexType>
    <xs:all minOccurs = 1>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

Assignment Project Exam Help

<https://powcoder.com>

- Defines an element named "person" which must contain the "firstname" and the "lastname" elements. They can appear in any order but both elements MUST occur once and only once!
- If exists, maxOccurs must be 1, but minOccurs can be either 0 or 1
- With minOccurs="0", each element CAN appear zero or one time!

Add WeChat powcoder

e.g.) <person>
 <firstname>Kimberly</firstname>
 </person>
is NOT valid.

Example: xs:choice

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="member" type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

- Defines an element named "person" which must contain either a "employee" element or a "member" element, not both.
- minOccurs and maxOccurs can be defined per element.

Example

- Persons.xsd
- Persons.xml

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

xs:attribute

- xs:attribute elements can be used within a complex type to indicate attributes of elements of that type.
- attributes of xs:attribute:
 - name
 - type
 - use = "required" or "optional".

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

With xs:attribute

```
<xs:complexType name = "movieType">  
  <xs:sequence>  
    <xs:attribute name = "title" type = "xs:string" />  
    <xs:attribute name = "year" type = "xs:integer"/>  
  </xs:sequence>  
</xs:complexType>
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

With sub-elements

```
<xs:complexType name = "movieType" >  
  <xs:sequence>  
    <xs:element name = "Title " type = "xs:string" />  
    <xs:element name = "Year" type = "xs:integer" />  
  </xs:sequence>  
</xs:complexType >
```

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

Example

- MoviesWithAttribute.xsd
 - MoviesWithAttribute.dtd
 - MoviesWithAttribute.xml
- Assignment Project Exam Help
<https://powcoder.com>

Add WeChat powcoder

Restricted Simple Type

- Restricted simple type can be the type of elements or attributes.
- `xs:simpleType` can describe enumerations and range-restricted base types.
- `name` is an attribute
- `xs:restriction` is a sub-element.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

<xs:restriction>

- Attribute base gives the simple type to be restricted, e.g., xs:integer.
- Subelements
 - xs:{min, max}{Inclusive, Exclusive} are four attributes that can give a lower or upper bound on a numerical range.
 - or
 - xs:enumeration is a subelement with attribute value that allows enumerated types.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example (a)

```
<xs:simpleType name = "movieYearType">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="1915"/>  
    <xs:maxInclusive value="2013"/>  
  </xs:restriction>  
</xs:simpleType>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example (b)

```
<xs:simpleType name = "movieGenreType">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value = "comedy" />  
    <xs:enumeration value = "drama"/>  
    <xs:enumeration value = "sciFi"/>  
    <xs:enumeration value = "teen"/>  
  </xs:restriction>  
</xs:simpleType>
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Example

- MoviesWithSimpleType.xml
 - MoviesWithSimpleType.xsd
- Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Keys in XML Schema

- An **xs:element** can have an **xs:key** subelement.

`<xs:element name = element name>`

`<xs:key name = key name>`

`<xs:selector xpath = "path description" />`

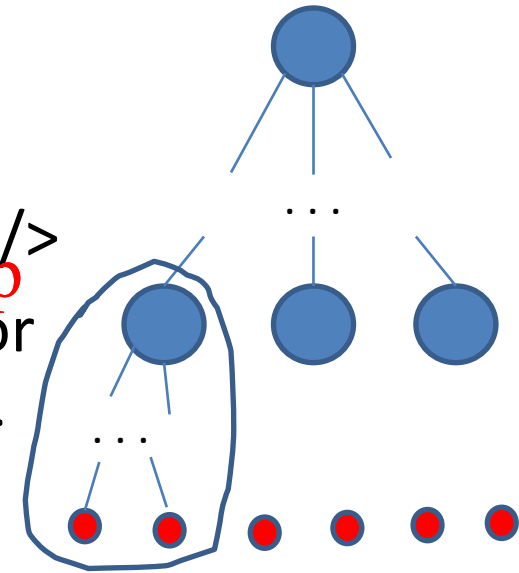
`<xs:field xpath = "path description" />` or

`<xs:field xpath = "@path description" />`

`</xs:key>`

`</xs:element>`

- The key element MUST contain the following (in order):
 - one and only one selector element
 - one or more field elements to form a key. The field can be any sub element of the last element on the selector path or an attribute of the last element.



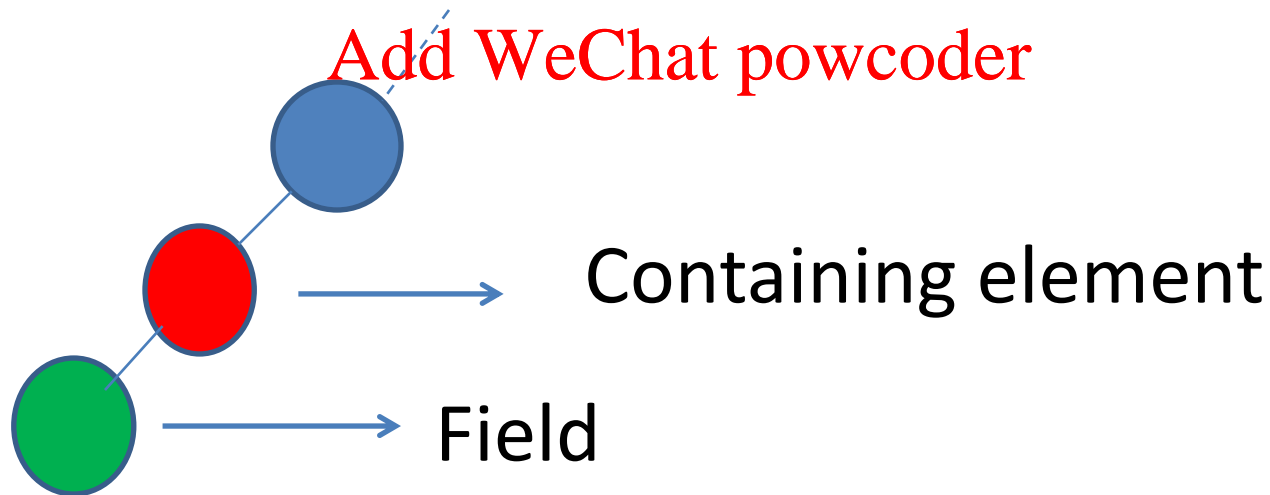
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Keys in XML Schema

- Selector: Xpath to the containing element
- Field: Xpath to an attribute or element of which value (or set of values) must be a key within the containing element.



Example

```
<xs:element name = "Movies">
  <xs:complexType>
    <xs:sequence>
      <xs:element name = "Movie" type = "movieType" minOccurs = "0" maxOccurs =
"unbounded" />
      <xs:element name = "MovieSeries" type = "seriesType" minOccurs = "0"
maxOccurs = "unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:key name = "movieKey">
    <xs:selector xpath = "Movie" />
    <xs:field xpath = "Title" />
    <xs:field xpath = "Year" />
  </xs:key>
</xs:element>
```

Assignment Project Exam Help

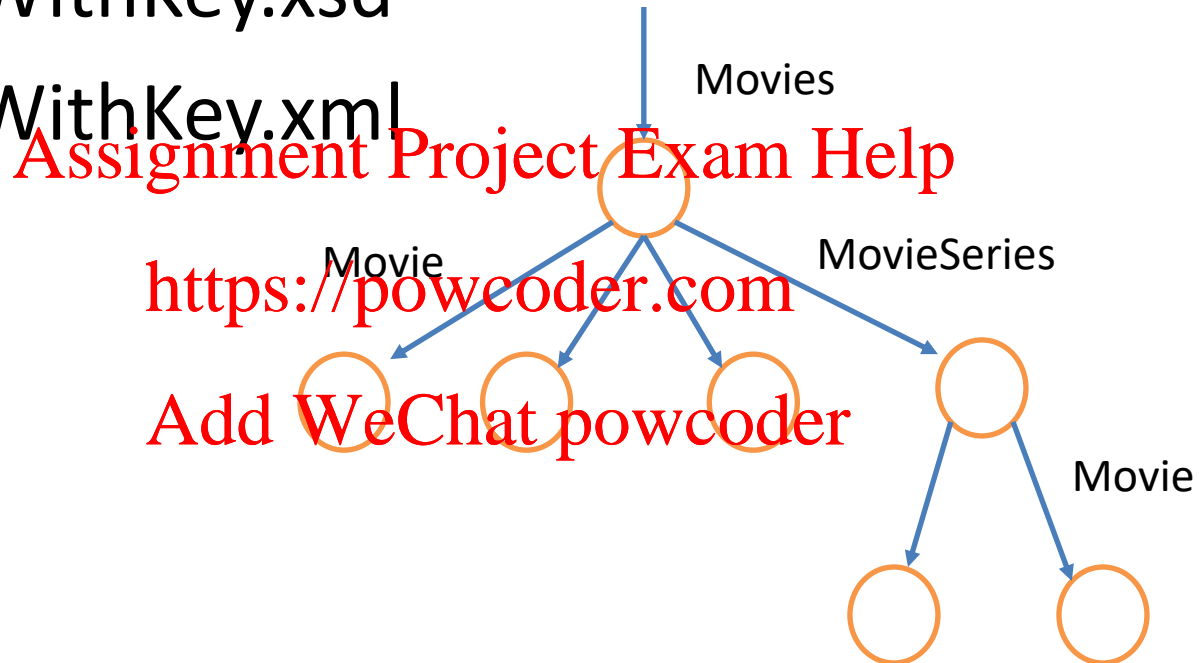
<https://powcoder.com>

Add WeChat powcoder

Note: The key name "movieKey" will be used if it is restricted by a foreign key.

Example

- MoviesWithKey.xsd
- MoviesWithKey.xml



xs:key vs xs:unique

- xs:key

The field must exist.

Assignment Project Exam Help

- xs:unique

<https://powcoder.com>

The field might not exist, and the constraint is

Add WeChat powcoder

only that they are unique if they exist.

| | Relational Model | XML |
|-----------|---------------------------|-------------------------------------|
| Structure | Tables | Hierarchical Tree |
| Schema | Fixed in advance required | Flexible "self-describing" optional |
| Queries | SQL | XPath, XQuery, XSLT |
| Ordering | None | Implied ordering |

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder