# Assignment # 1 – XML Schema

**Due Date:** June 4th, 2012, 6:00 PM, Tuesday class

# Objectives

Learn XML Schema grammar and create a schema demonstrating a specified set of rules to validate against.

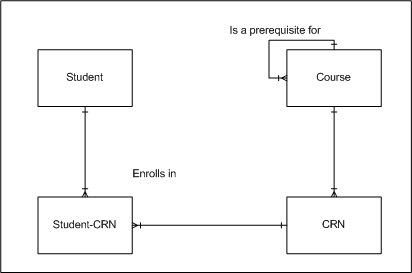
# Description

XML and DTD validation has helped to demonstrate diving into data modeling and validation within the world of XML. However, some of the limitations of DTD have restrained you from coming up with more complex rules for validation within XML. This assignment focuses on using XML Schema to validate your XML.

# Requirements Build an XML Schema to restrict what goes into the document. The data model and types are:

* A Student which contains:
  + email addresses (up to and including 4) and follow an email address regular expression pattern
  + first name (not allow numeric values – only alphabetic)
  + ID, can be an attribute or an element and must follow the BCIT naming convention (e.g. A12345678)
  + last name (not allow numeric values – only alphabetic)
  + three phone numbers: home, work, and optional cell, each phone number must have 3 parts: area code exchange code and extension code
  + address which contains sub-parts: optional suite, address number, street, city, postal code (conforms to a regular expression that restricts values to Canadian postal code format), could be attributes or elements
* A Course which contains:
  + credits
  + description
  + number, can be an attribute or an element
  + name
  + a grouped list of prerequisite courses where the prerequisite courses are references to other courses.
* CRN which contains:
  + ID, can be an attribute or an element
  + course number, can be an attribute or an element
  + total number of students
  + end date (of type date)
  + start date (of type date)
  + tuition (currency so should be limited to two decimal places)
  + instructor name (not allow numeric values – only alphabetic)
* Student CRN which contains
  + student ID
  + CRN

The entity relationships are shown in figure 1.



Figure

# Additional Details

• Use element pools in order to put all of these different types of elements into one XML document

* + So have a school root element that contains:
    - A courses element
    - A students element
    - An CRNs element
    - A student-CRNs element
  + Where each of the above elements contains their respective child type elements (e.g. customers element contains customer elements, etc.)
* Create at least three records inside of each of the courses, students, CRNs, student-CRNs elements

## Keys Instead of ID/IDREF

Use keys and keyrefs for your unique identifiers. You will need to create keys for the following:

* Student (based on id)
* Course (based on number)
* CRN (base on crn)
* student-CRN (based on both the studentID and the crn)

## Content

Any data type that gets reused for multiple elements must be **global**. You must also **demonstrate three example records of each of the entities found in figure 1**. Both your XML Schema must be well-formed and conform to the XML Schema recommendation and your XML document must be well-formed and validate against your XML Schema.

Create two documents:

1. A XML Schema (.xsd) with rules (from above)
2. An XML document that conforms to those rules

# Hints

* Refer to the web sites found at the back of slides for week four and five
* Use the examples that are posted on share out. Many of them demonstrate very similar (or even the same) data models and types as what you are being ask to do for this assignment
* Validate often and save your work (use the example code posted to share out)
* Some of the child elements can be based on a similar derived type (e.g. first, middle, and last name)
* Think of when you want to create simple types and complex types
* Think about whether you want to reference or derive a type
* When attempting to perform the cardinality, you will need two keys and one key reference. The first key is for the ID of the employee and the other is to enforce cardinality
* You may wish to slightly altar the dependants element pool to further support your cardinality rule

# Submission

You will work individually for this assignment.

Your submission will be **one zip file** which contains 1 XML file with the extension being “.xml” and one XML schema file with the extension “.xsd”. The naming convention will be lastname-firstname-assign4.xml (e.g. Ferguson-Arron-assign4.xml) and submitted to D2L.

# Marks

The assignment is worth 10 marks in total.

* **2 marks** – XML Schema validates & is well-formed
* **2 marks** – XML document validates against your XML Schema and is well-formed
* **6 marks** – For the rest of the requirements found in this document
  + **Up to -5 marks off** if the XML Schema does not validate against the XSD rules & is not well-formed
  + **Up to -5 marks off** if the XML document does not validate against your XML Schema and is not well-formed