

XML und Datenbanken

— 5. Übungsblatt: XML Schema, Teil 2 —

Allgemeine Aufgabe

Notieren Sie sich eventuelle Verständnisfragen, so dass wir diese im nächsten Online-Treffen klären können. Da es erfahrungsgemäß sehr sill ist: Nehmen Sie sich bitte die Zeit und denken Sie bewusst über mögliche Fragen nach (wenigstens eine). Ich kann auch Studierende drannehmen, die sich nicht gemeldet haben. Ein Mal würde ich wohl verstehen, dass der Stoff so einfach war, dass Sie einfach keine Fragen haben. Wenn sich das wiederholt, müsste ich dann aber umgekehrt prüfungsähnliche Fragen stellen.

Hausaufgabe

Geben Sie die folgenden Aufgaben bis Montag, 21.11.2022, 16⁰⁰, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösung in eine `.xml`-Datei. Eventuelle Anmerkungen markieren Sie bitte als XML-Kommentar. Es gibt 2 Punkte für Teil a), 6 Punkte für Teil b) und je 2 Punkte für Teil c) und d). Aufgabe e) ist eine Bonusaufgabe. Sie bekommen 4 Punkte, wenn Sie sie lösen.

Die Abgaben nur stichprobenartig kontrolliert. Wenn Ihre Abgabe nicht kontrolliert wurde, bekommen Sie die volle Punktzahl. Wenn Sie später wegen Plagiaten auffallen, oder bei einer Stichprobe eine fast gar nicht gelöste Aufgabe entdeckt wird, können auch alte Abgaben kontrolliert werden. Dann können auch rückwirkend Punkte abgezogen werden. Sie benötigen 67% der Hausaufgabenpunkte und eine aktive Mitarbeit in den Übungen für die Studienleistung.

Die „Wiederholungsaufgaben“, also Teil f) bis j), sind nicht abzugeben. Beschäftigen Sie sich aber bitte auch mit diesen Aufgaben. Sie müssen damit rechnen, dass Sie beim Online-Treffen gebeten werden, einen Teil des Vorlesungs-Stoffes zu wiederholen und insbesondere eine der Fragen aus f) zu beantworten.

- a) Überarbeiten Sie nochmals die Struktur Ihrer XML-Datei für Einkäufe, die Sie schon in den Hausaufgaben 1b), 2a) und 4a) erstellt haben. Diesmal ändern Sie bitte etwas: Wenn Sie bisher hauptsächlich Attribute verwendet haben, nutzen Sie jetzt vorwiegend Elemente mit einfachem Inhalt (und umgekehrt). Ziel ist es, dass Sie alle Konstrukte einmal kennenlernen (inklusive der Definition in XML Schema). Es ist jetzt auch erlaubt, die Anwendung etwas zu verbessern, wenn Sie mögen. Sie müssen sich also nicht exakt an das vorgegebene relationale Schema halten.

Z.B. gibt es Waren, die nach Stück verkauft werden, und solche, die nach Gewicht verkauft werden. Bei denen, die nach Stück verkauft werden, ist das Gewicht aber öfters auch bekannt (z.B. war der Kauf von „Trauben hell“ eine Schale von 400g, eigentlich also ein Stück). Falls Sie keine Idee für eine gute Lösung haben, können Sie die Daten natürlich auch so lassen.

Sie können das Datum im Format des entsprechenden „Simple Type“ von XML Schema speichern. Sie müssen das nicht tun, dann sollten Sie die möglichen Werte aber mit einem regulären Ausdruck so einschränken, dass zumindest die grobe Struktur eines Datums erzwungen wird (nicht unbedingt auch die Anzahl Tage im Monat).

- Zu jeder Ware ist ein eindeutiges Kürzel (eine ID), sowie Name und eine Kategorie abzuspeichern.

WAREN		
ID	NAME	KATEGORIE
WM	Weidemilch 3.5%	Milch
KZB	Käse-Zwiebel-Brötchen	Brot
BAN	Bananen	Obst
TRH	Trauben hell	Obst

- Zu jedem Einkauf ist das Datum und der Laden zu speichern.

EINKAUF		
NR	DATUM	LADEN
1	11.10.2022	Aldi
2	15.10.2022	NP

- Außerdem ist für jeden Einkauf zu speichern, welche Waren Sie in welcher Menge zu welchem Preis gekauft haben.

GEKAUFT				
NR	ID	PREIS	MENGE	BEM
1	WM	1.45	1	
1	KZB	0.49	2	
1	BAN	1.00	0.5	
1	TRH	4.98	0.4	
2	BAN	1.49	1.2	WWF-Bananen

- b) Entwickeln Sie ein XML Schema für XML-Dateien, die so wie in Ihrer Lösung von Aufgabe a) strukturiert sind. Wenn Ihr Schema zu Aufgabe 4b) nicht im „Venetian Blind“-Stil war, wählen Sie jetzt diesen Stil. Falls Sie den „Venetian Blind“-Stil schon verwendet hatten, wählen Sie nun den „Salami Slice“-Stil.

Definieren Sie Schlüssel und Fremdschlüssel. Verwenden Sie auch Einschränkungen für „Simple Types“, z.B. sollten viele Zahlwerte nicht negativ sein. Die möglichen Kategorien von Lebensmitteln definieren Sie als Aufzählung.

Stellen Sie unbedingt sicher, dass Ihre Daten-Datei aus a) bezüglich Ihres Schemas aus b) auch validiert werden kann.

Da wir nur einzelne Lösungen bei dem Online-Treffen besprechen können, ist wirklich wichtig, dass Sie sich selbst durch Anwendung eines Validators das notwendige Feedback beschaffen. Lösungen, die offensichtlich nicht validiert wurden, würde ich als Täuschungsversuch verstehen.

Mögliche Validatoren sind z.B.

- [<https://www.xmlvalidation.com/>]
(Wenn Sie „Validate against external XML schema“ anklicken, können Sie im nächsten Schritt das Schema eingeben.)
- [<http://www.freeformatter.com/xml-validator-xsd.html>]

Bei Schwierigkeiten können Sie das Forum benutzen oder mir eine EMail schicken.

Es ist auch zu empfehlen, zu Testzwecken Fehler in die Daten-Datei einzubauen, und zu prüfen, dass die Datei dann tatsächlich nicht validiert werden kann.

- c) Definieren Sie einen Datentyp `length` für Längenangaben (mit Einheit). Es soll eine Zeichenkette sein, die zuerst eine Folge von Ziffern enthält, und dann genau einen der Suffixe `mm`, `cm`, `m`. Führende Nullen sollen nicht erlaubt sein, aber die Länge 0 (mit einer der Einheiten) muss möglich sein. Schreiben Sie ein minimales Schema und XML Dokument, um Ihre Definition zu testen. Geben Sie das Schema ab mit dem Testdokument als Kommentar.
- d) Definieren Sie einen Datentyp für Klausurergebnisse: Er soll Zahlwerte von 0 bis 100 erlauben (ganze Zahlen, die als Prozentwerte gedacht sind), und alternativ die Angabe `NE` (nicht erschienen). Testen Sie Ihre Definition wieder wie in c).
- e) Dies ist eine optionale Bonusaufgabe (die Punkte werden bei der zu erreichenden Gesamtpunktzahl nicht mitgezählt).

Schreiben Sie einen Test für Schlüssel und Fremdschlüssel in einem komplizierteren Fall (Sie können sich aussuchen, welchen):

- Der referenzierte Schlüssel ist in einem Elternelement des Fremdschlüssels definiert.

- Der referenzierte Schlüssel ist in Vorgängern des Fremdschlüssels definiert. Es gibt mehrere solche Vorgänger mit teils überlappenden Schlüsselwerten. Sie referenzieren einen eindeutigen Wert, der aber nicht im nächstliegenden Vorgänger definiert ist.
- Ebenso, aber Referenzierung eines nicht eindeutigen Wertes.
- Der referenzierte Schlüssel ist in einem Kindelement des Fremdschlüssels definiert. Es gibt genau ein passendes Kind.
- Der referenzierte Schlüssel ist in einem Kindelement des Fremdschlüssels definiert. Es gibt mehrere passende Kindelemente (also mehrere Instanzen des Schlüssels), aber die Schlüsselwerte sind disjunkt.
- Ebenso, aber mit nicht disjunktion Schlüsselwerten. Sie müssen natürlich einen eindeutigen Wert referenzieren.
- Sie können auch noch weitere Varianten testen, die Sie interessant finden.

Es ist natürlich in Ordnung, wenn einige der Tests dazu führen, dass das Dokument nicht validiert werden kann. Ziel ist, unsere Erkenntnisse zusammenzutragen. Sie können gerne auch mehrere verschiedene Validatoren ausprobieren.

Prüfen Sie, ob die Ergebnisse Ihrer Tests kompatibel mit den Angaben im Skript sind.

Wiederholungsaufgaben

Beschäftigen Sie sich mit diesen Aufgaben. Sie brauchen aber nichts abzugeben.

- f) Was würden Sie in einer mündlichen Prüfung auf folgende Fragen zu XML Schema antworten?
- Was ist das Format für einen `dateTime` Wert in XML schema? Wie definieren Sie einen Wert für den “22. November 2018, 9:00 MEZ” (Mittleuropäische Zeit, eine Stunde später als UTC)?
 - Ist ein Listen-Typ ein einfacher Typ oder ein komplexer Typ in XML Schema? Angenommen, Sie haben eine Liste von ganzen Zahlen als Typ definiert. Wie würde eine lexikalische Repräsentation aussehen?
 - Geben Sie ein Beispiel für eine Anwendung eines `union`-Typs.
 - Nennen Sie ein Werkzeug, das Sie zum Validieren benutzt haben.
 - Wie spezifiziert man Schlüssel in XML Schema?
 - Was unterscheidet Schlüssel für XML von Schlüsseln in relationalen Datenbanken?
 - Wie spezifiziert man Fremdschlüssel in XML Schema?
 - Welche Vor- und Nachteile haben ID/IDREF im Vergleich zu Schlüsseln und Fremdschlüsseln?
 - Macht es einen Unterschied, ob Sie in einer Element-Deklaration keinen Typ definieren, also das `complexType`-Element weglassen, oder ein leeres `complexType`-Element hinschreiben?

Für Interessierte

- g) Informieren Sie sich über Sitemaps für Google im XML Format:

[https://support.google.com/webmasters/answer/183668?hl=en&ref_topic=4581190]

Weitere Informationen gibt es unter

[<http://www.sitemaps.org/protocol.html>]

Lesen Sie das Schema unter

[<http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd>]

Entwickeln Sie eine Sitemap (XML-Datei) mit mindestens drei URIs, die bezüglich dieses Schemas validiert werden kann. Testen Sie die Validierung. Sie könnten z.B. eine Sitemap für die Webseite der Vorlesung und einige der Unterseiten entwickeln, können aber auch eine eigene Webpräsenz als Beispiel nutzen, oder ein fiktives Beispiel.

- h) Wenn Sie wollen, können Sie auch die Google Erweiterung für Image Sitemaps anschauen (und in Ihrem Beispiel nutzen):

[<https://support.google.com/webmasters/answer/178636>]

Eine XML Schema Datei dazu finden Sie hier:

[<https://www.google.com/schemas/sitemap-image/1.1/sitemap-image.xsd>]

- i) Falls Sie sich weitere bekannte XML Schema Dateien anschauen wollen, gibt es u.a. folgende Möglichkeiten. Natürlich brauchen Sie die jeweilige Technologie inhaltlich nicht zu verstehen. Aber Sie könnten diese XML Schemata einmal durchscrollen und sehen, dass Sie die dort verwendeten Konstrukte schon kennen.

- Ein XML Schema für XHTML finden Sie hier:

[<https://www.w3.org/TR/xhtml1-schema/>]

- XML Schema for SEPA Lastschriften (zur Einreichung bei einer Bank):

[<https://www.europeanpaymentscouncil.eu/document-library/implementation-guidelines/sepa-credit-transfer-scheme-customer-bank-implementation>]

Weitere Informationen finden Sie hier:

[<https://www.hettwer-beratung.de/sepa-spezialwissen/sepa-technische-anforderungen/xml-sepa-schema-xsd-dateiformat/>]

- j) Für reguläre Ausdrücke sei auf diese Seite verwiesen:

[<https://regexcrossword.com/>]