Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

08. November 2018

XML und Datenbanken
— 3. Übungsblatt: Vom ER-Modell zu DTDs —

Bei Teil a) bis d) handelt es sich um Präsenzübungen, die nicht abgegeben werden müssen. Sie sollten über die Wiederholungsfragen a) und b) allerdings vor der nächsten Übung selbst nachdenken. Teil e) ist eine Hausaufgabe. Bitte schicken Sie eine Lösung per EMail an den Dozenten (mit "`xml18`" in der Betreff-Zeile, bis zum 14.11.2018). Schicken Sie bitte keine leeren EMails nur mit Anhang, diese landen im Spam-Ordner.

a) Wie würden Sie in einer mündlichen Prüfung auf folgende Fragen zu Namespaces antworten?

- Warum wurden XML Namespaces eingeführt? Welches Problem sollen sie lösen? Nennen Sie mindestens ein typisches Beispiel, wo dieses Problem auftritt.
- Wie deklariert einen Präfix für einen Namespace (URI) im XML Dokument? Was ist der Gültigkeitsbereich dieser Deklaration?
- Wie deklariert man einen Default-Namespace?
- Woran erkennt eine XML-Anwendung, die Namespaces verwendet, "ihre" Elemente ("Tags")? Wenn man z.B. den Präfix "`xsd`" durch "`abc`" ersetzt, würde das auch funktionieren? Wenn man die Groß-/Kleinschreibung in der Namespace URI ändert (im Domain-Anteil), würde das einen Unterschied machen?

b) Was würden Sie in einer mündlichen Prüfung auf folgende Fragen zur Übersetzung eines relationalen Schemas bzw. eines ER-Diagramms in eine XML DTD antworten?

- Welche Möglichkeiten gibt es, eine Tabelle einer relationalen Datenbank nach XML zu übersetzen?
- Elemente in XML Dokumenten haben ja immer eine definierte Reihenfolge, während die Zeilen einer Tabelle das nicht haben. Gibt es Situationen, wo man deswegen bei der Übersetzung etwas einsparen kann?
- Was sind die Probleme bei Schlüsseln und Fremdschlüsseln im Vergleich zu `ID/IDREF`?
- Welche Übersetzungsmöglichkeiten gibt es für eins-zu-viele Beziehungen? Nennen Sie mindestens zwei Möglichkeiten. Diskutieren Sie ggf. Einschränkungen und Probleme.
- Welche Übersetzungsmöglichkeiten gibt es für viele-zu-viele Beziehungen?
- Wie kann man vorgehen, wenn ein beliebiges ER-Diagramm gegeben ist?

Präsenzaufgaben

c) Falls Sie in Ihrer Hausaufgabe nicht `ID/IDREF`, `NMTOKEN` und einen Aufzählungstyp für die Anrede ("Herr", "Frau") verwendet haben, ändern Sie Ihre Lösung bitte entsprechend. Wenn Sie das nicht als Verbesserung empfinden, lassen Sie uns darüber diskutieren. Es ist sehr wichtig, dass Sie alle Lösungen validieren lassen, z.B. mit

[https://www.truugo.com/xml_validator/]

oder

[http://www.xmlvalidation.com/?L=2]

Probieren Sie auch aus, was passiert, wenn Sie absichtlich Fehler in Ihr Dokument einbauen. Prüfen Sie auch, ob BaseX diese Fehler findet (sowohl mit der DTD im Dokument selbst, als auch mit Verweis aus dem XML Dokument auf die DTD). Wie verhält sich BaseX, wenn Elemente in der DTD fehlen?

d) Es sei die folgende relationale Datenbank gegeben. Entwickeln Sie eine DTD für eine Repräsentation dieser Daten in XML.

Es könnte nützlich sein, wie in der Vorlesung für eins-zu-viele Beziehungen beschrieben, sich einen Graphen aufzuzeichen mit den Relationen als Knoten und den Fremdschlüssel-Beziehungen als Kanten (in umgekehrter Richtung). Da Sie mit Hilfe von Schachtelung nur Bäume darstellen können, müssen Sie einige Kanten aus dem Graphen streichen und diese Kanten dann mit `ID/IDREF` oder "Fremdschlüssel"-Attributen darstellen.

Durchdenken Sie auch eventuelle Einschränkungen Ihrer DTD. Welche Integritätsbedingungen müssten zusätzlich überwacht werden? Versuchen Sie von den Möglichkeiten des Datenmodells (XML mit DTDs) guten Gebrauch zu machen, so dass nur wenig zusätzlich geprüft werden muss.

Die Datenbank ist Teil eines Systemkatalogs einer relationalen Datenbank. Sie enthält die folgenden Tabellen:

- `USERS(`<u>`USER`</u>`, FULLNAME°)`
  Diese Tabelle enthält einen Eintrag für jeden Nutzer der Datenbank.

USERS	
<u>USER</u>	FULLNAME
SYS	Data Dictionary Owner
SB	Stefan Brass

- `TABS(`<u>`OID`</u>`, OWNER→USERS, TABNAME)`
  Die zweite Tabelle enthält einen Eintrag für jede Tabelle, die im System gespeichert ist. Tabellen werden durch eine "Object ID" identifiziert, und alternativ durch den Besitzer der Tabelle und den Tabellen-Namen (wie in Oracle hat jeder Nutzer sein eigenes Schema).

TABS		
OID	OWNER	TABNAME
1	SYS	USERS
2	SYS	TABS
3	SYS	COLS
4	SYS	GRANTS
5	SB	STUDENTS

- `COLS(OID→TABS, COLNO, COLNAME)`
  Diese Tabelle enthält einen Eintrag pro Spalte. Die Einträge sind identifiziert durch die `OID` der Tabelle und die Nummer der Spalte in der Tabelle.

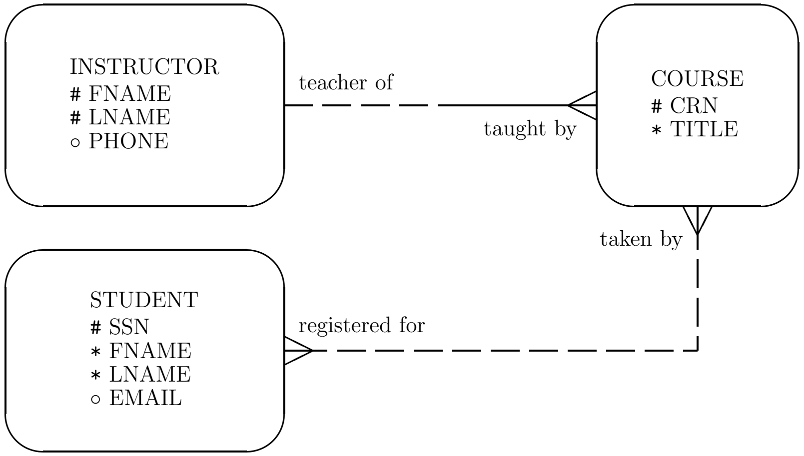
COLS		
OID	COLNO	COLNAME
1	1	USER
1	2	FULLNAME
⋮	⋮	⋮

- `GRANTS(OID→TABS, GRANTOR→USERS, GRANTEE→USERS, PRIV, GRANTABLE)`
  Diese Tabelle enthält die Zugriffsrechte. Es wird gespeicher wer (`GRANTOR`) wem (`GRANTEE`) ein Zugriffsrecht (`PRIV`) an welcher Tabelle (`OID`) gegeben hat, und ob der Nutzer das Recht weitergeben darf (`GRANTABLE`). Dies entspricht in SQL: "`WITH GRANT OPTION`".

GRANTS				
OID	GRANTOR	GRANTEE	PRIV	GRANTABLE
1	SYS	BRASS	SELECT	N
⋮	⋮	⋮	⋮	⋮

Hausaufgabe

e) Geben Sie zwei deutlich verschiedene Übersetzungen des folgenden ER-Diagramms in eine XML DTD an. Sie dürfen bei Bedarf weitere ID-Attribute hinzufügen. Diskutieren Sie kurz die Vor- und Nachteile jeder Lösung. Insbesondere nennen Sie bitte ggf. zusätzliche Bedingungen, die überprüft werden müssten, damit Ihre DTD nur Daten zulässt, die dem ER-Diagramm entsprechen.



Entwickeln Sie zu Ihren DTDs jeweils auch eine kurze XML-Datei und validieren Sie diese. Sie können z.B. die folgenden Daten verwenden:

INSTRUCTOR		
FNAME	LNAME	PHONE
Stefan	Brass	24750
Stefan	Posch	

COURSE			
CRN	TITLE	INST_FNAME	INST_LNAME
1	XML und Datenbanken	Stefan	Brass
2	Objektorientierte Programmierung	Stefan	Brass
3	Bildverarbeitung	Stefan	Posch

STUDENT			
SSN	FNAME	LNAME	EMAIL
101	Lisa	Weiss	weiss@acm.org
102	Daniel	Sommer	

REGISTERED_FOR	
SSN	CRN
101	1
101	3
102	1