Module 4: XML Storage

The major aspects of storing XML include

- XML Keys
- Concepts: Data and Document Centrism
- Storage
- Mapping to relational schemas
- SQL/XML
Integrity Constraints in XML

XML Schema

- Entity: xsd:unique and xsd:key
- Referential: xsd:keyref
- Data type: XML Schema specifications
  - Value: Solve custom queries using XPath or XQuery

Entity and referential constraints are based on XPath
XML Keys: 1

Keys serve as generalized identifiers, and are captured via XML Schema elements:

- **Unique**: candidate key
  - The selected elements yield unique field tuples

- **Key**: primary key, which means candidate key plus
  - The tuples exist for each selected element

- **Keyref**: foreign key
  - Each tuple of fields of a selected element corresponds to an element in the referenced key
XML Keys: 2

Two subelements built using restricted application of XPath from within XML Schema

- **Selector**: specify a set of objects: this is the scope over which uniqueness applies
- **Field**: specify what is unique for each member of the above set: this is the identifier within the targeted scope
  - Multiple fields are treated as ordered to produce a tuple of values for each member of the set
  - The order matters for matching keyref to key
Selector XPath Expression

A selector finds descendant elements of the context node

- The sublanguage of XPath used allows
  - Children via ./child or .//* or child
  - Descendants via .// (not within a path)
  - Choice via |

- The subset of XPath used does not allow
  - Parents or ancestors
  - text()
  - Attributes
  - Fancy axes such as preceding, preceding-sibling, ...
Field XPath Expression

A field finds a unique descendant element (simple type only) or attribute of the context node

- The subset of XPath used *allows*
  - Children via ./child or ./*
  - Descendants via .// (not within a path)
  - Choice via |
  - Attributes via @attribute or @*

- The subset of XPath used *does not allow*
  - Parents or ancestors
  - text()
  - Fancy axes such as preceding, ...

An element yields its text()
XML Foreign Keys

```xml
<keyref name="..." refer="primary-key-name">
  <selector xpath="..."/>
  <field name="..."/>
</keyref>
```

- Relational requirement: foreign keys don’t have to be unique or non-null, but if one component is null, then all components must be null.
Placing Keys in Schemas

- Keys are associated with elements, not with types
- Thus the . in a key selector expression is bound
- Could have been (but are not) associated with types where the . could be bound to whichever element was an instance of the type

&text() is implicit
<key . . . .
  <selector - same >
  <field @name ti />
  <field @lg />

field  concat (@tjelb)

abc def
abcd ef

English Art Glisten en
<choice>
  or
  two.

selector → node set

one/two

field

uniqueness