Course NPRG039: **Advanced Aspects and New Trends in XML**

Lecture 06:

Native XML Databases – Part 2

Martin Svoboda

1. 12. 2015

Charles University in Prague, Faculty of Mathematics and Physics

Outline

- Native XML databases
  - Oracle Berkeley DB XML
  - Timber
  - **Sedna**

Sedna

Introduction

- **Sedna**
  - http://www.sedna.org/
  - Free native XML database system
    - Open source under Apache License 2.0
    - Implemented in C/C++
    - Drivers for Java, C, PHP, Python, Ruby, Perl, C#, …
  - Platforms
    - Linux, Windows, Mac OS X, FreeBSD, Solaris

Introduction

- Basic features
  - **XQuery** language
    - SQL connection from XQuery
    - External functions implemented in C
  - Declarative node-level **updates**
  - **Indices** based on B-trees, **full-text** search indices
  - ACID **transactions**
  - Triggers
  - Incremental backup
  - Database security (users, roles, authentication)

Documents

- Document creation and removal
  - **`CREATE DOCUMENT`** `name`
  - **`DROP DOCUMENT`** `name`
- Bulk load
  - **`LOAD`** `location name`
    - Loads the specified file into a given standalone document
- Document retrieval
  - **`doc`**`($name)`

Collections

- **Collections**
  - Documents can be organized into collections
    - However, standalone documents are permitted as well
    - I.e. each document belongs to at most one collection
- Management of collections
  - **`CREATE COLLECTION`** `name`
  - **`DROP COLLECTION`** `name`
  - **`RENAME COLLECTION`** `old-name` **`TO`** `new-name`

Collections

- Document creation and removal
  - **`CREATE DOCUMENT`** `name` **`IN COLLECTION`** `col`
  - **`DROP DOCUMENT`** `name` **`IN COLLECTION`** `col`
- Bulk load
  - **`LOAD`** `location name col`
- Document retrieval
  - **`doc`**`($name, $col)`
    - Returns a given document from the specified collection
  - **`fn:collection`**`($col)`
    - Returns a sequence of all documents of a given collection

Updates

- Insert
  - **`UPDATE INSERT`** `source` (**`INTO`** | **`PRECEDING`** | **`FOLLOWING`**) `target`
    - Inserts zero or more nodes into a designated position
- Delete
  - **`UPDATE DELETE`** `expression`
    - Removes target nodes including their descendants

Updates

- Replace
  - **`UPDATE REPLACE`** `$var` **`IN`** `source` **`WITH`** `expression`
    - Replaces nodes with a new sequence of zero or more nodes
- Rename
  - **`UPDATE RENAME`** `target` **`ON`** `name`
    - Changes a name of target nodes

Value Indices

- Index creation
  - **`CREATE INDEX`** `Title`
    `ON` `path1` **`BY`** `path2` **`AS`** `type`
    - title – unique name of the index to be created
    - path1 – absolute path to nodes to be indexed
    - path2 – relative path to keys of such nodes
    - type – atomic type to which keys should be cast
- Index removal
  - **`DROP INDEX`** `title`

Value Indices

- Index usage
  - Existing indices are not exploited automatically!
    - Special functions have to be used instead
  - **`index-scan`**`($title, $value, $mode)`
    - Traverses the given index and returns a sequence of nodes matching the provided atomic value
    - $title – name of the index to be used
    - $mode – 'EQ', 'LT', 'GT', 'GE', 'LE'
  - **`index-scan-between`**`($title, $value1, $value2, $mode)`

Full-Text Indices

- Index creation
  - **`CREATE FULL-TEXT INDEX`** `title`
    `ON` `path` **`TYPE`** `type`
    `WITH OPTIONS` `options`
    - title – unique name of the index to be created
    - path – absolute path to nodes to be indexed
    - type – how to construct text representations of nodes
      - "xml" – XML representation
      - "string-value" – concatenation of all descendant text nodes
      - "customized-value"
    - options
      - Backend index implementation, stemming language, …

Full-Text Indices

- Index removal
  - **`DROP FULL-TEXT INDEX`** `title`
- Index usage
  - **`ftindex-scan`**`($title, $query, $options)`
    - Scans the given full-text index and returns a sequence of items which satisfy the query
    - Allowed query constructs:
      - Phrases – e.g. `'word1 word2'`
      - Operators – e.g. `'word1 OR word2'`
      - Stemming – e.g. `'word~'`
      - Contains – e.g. `'element CONTAINS word'`

Metadata

- **Metadata**
  - Accessible as special system documents / collections
    - They can be queried, but not updated
    - Their names start with $
  - `$documents`
    - Document with a list of all documents
  - `$collections`
    - Document with a list of all collections
  - `$indexes`
    - Document with a list of all indexes
  - …

Java API

- **Sessions**
  - `SednaConnection c = DatabaseManager.getConnection( url, dbname, user, password );`
    - Establishes an authenticated connection with the server
      - url – server name including an optional port (default 5050)

Java API

- **Statements**
  - `SednaStatement s = c.createStatement();`
    - Creates a new statement object
  - `s.loadDocument(inputStream, name);`
    `s.loadDocument(inputStream, name, col);`
    - Loads contents of a given standalone / collection document
  - `s.execute(query);`
    - Executes a given query statement

Java API

- **Results**
  - `SednaSerializedResult r = s.getSerializedResult();`
    - Retrieves the result of the last executed query
  - `r.next();`
    - Fetches the next item from the result
    - Otherwise `null` is returned

Java API

- **Transactions**
  - `c.begin();`
  - `c.commit();`
  - `c.rollback();`
- **Exceptions**