

Theme 7

XML And Databases

Semistructured Data

- Data that may be **irregular** or **incomplete** and have a structure that may **change** rapidly or unpredictably
- Most semistructured data are based on query languages that traverse a **tree-labelled representation**.
- Because there is no fixed structure, data can only be obtained by **specifying its position** within the collection.
- Mean query loses its traditional **declarative** nature and become more **navigational**.

Database and XML

- There were once only two main options when deciding where to store your data...
- ...traditional solution, a **relational** database (Oracle, Microsoft SQL, MySQL)...
- ...the alternative solution, **XML**.

Relational databases

- Relational databases revolves around the idea to **recognize relations** among stored items of data
- They create these type of **relations** by use of **tables**, **keys**, and differ **type** of relationships
- Relationship such as **many-to-many** or **many-to-one**
- They can range from top-end database servers to ones designed for desktop uses

Relational databases

- Regardless, these system have two things in common:
 - **Structured Query Language (SQL)**
 - **They cope well with data that can be broken down**
- Yet, there are instances where the data does not fit easily into this patterns.
- Solutions: binary files or XML

Storing XML in relational databases

- There are four general approaches to **storing** an **XML document** in a relational database
 - Store the XML as the value of some **attribute**
 - Store the XML in a **shredded** form across a **number** of **attribute** and **relations**
 - Store the XML in a **schema-independent** form
 - Store the XML in a **parsed** form.

Problems with relational XML file system

- File structure
 - Naming convention and retrieval
- Two systems implementation
 - The database and XML processor
- Retrieving
 - Document vs fragments
- Indexing
 - Had to create and need another system to maintain
- Backup
 - One for the database and one for the XML files

Type of XML databases

- Solution – database that store XML and query them.
 - **XML enabled database** - DB stores data in non-XML (e.g. tables) but can convert its data to XML.
 - **Oracle Databases, PostgreSQL, Microsoft SQL Databases**
 - **native XML** database - DB stores data natively as XML.
 - **BaseX, Oracle Berkeley DB, eXist**
- Both database type includes features to address common task and issues.
- Dependant on the application, certain features will definitely be used more often than others

Native XML Database

- Defines a (logical) **model** for an XML document...
- ...as opposed to the data in that document...
- ...and stores and retrieves documents according to that model.
- At a **minimum**, the model must include elements, attributes, PCDATA, and document order.

Native XML Database

- Store XML documents as a unit...
- ...and will create a **model** that is closely aligned with **XML**...
- ...or one of XML's related technologies like the Infoset or DOM.
- Uses XML document as its fundamental unit of (logical) storage

Native XML Database

- Is not required to have any particular underlying physical storage model
- It can be built on a:
 - relational, hierarchical, or object-oriented database
 - or use a proprietary storage format such as indexed, compressed files.

XML Enabled Database

- A database that has an added XML mapping layer...
- ...provided either by the database vendor or a third party.

BUT

- Data that is mapped into the database is mapped into **application specific formats...**
- ...and the original XML **meta-data and structure may be lost.**
- Data retrieved as XML is **NOT** guaranteed to have originated in XML form.

Why XML Databases

- XML database will often far surpass a relational database in
 - convenience
 - ease of development
 - Performance
- XML databases are often used in:
 - Product catalogs
 - Patient information tracking
 - Corporate information portals
 - Membership databases
 - Document management/exchange systems

When to consider migrating to XML DB

- When records are represented as XML when they are processed or transmitted by applications
 - Underlying application uses XML as a means of communication
- When the data format (schema) changes over time
 - Changing existing structure within set databases are tedious
- When the data format (schema) is complex and highly variable
 - The complexity and variability and the number of optional data fields...
- When you need to optimize object-centric data access
 - Querying said relational databases

When to turn away...

- XML databases run slower
 - XML searches are slow
 - Difficulty with XML database conversion
 - XML limitations as a database
 - XML disadvantages in data security
-
- Other NoSQL solutions?

Mongog DB
DynamoDB
ElasticSearch

Recap...

Theme 7: XML & Databases

- TO BE CONTINUED...