Experiment No 7 - XML Programming

Aim : Query execution on XML database.

Write Ups :

Page No. 15
Date

Name : Kreena Shah
Sapid : 60004210243
Batch : Comps C'32
Subject : ADBMS
Date of Performance : 02/11/2023
Date of Submission : 02/11/2023

Aim : To Query XML databases.

Theory :

An XML database is a type of database management system specifically designed to store, retrieve & manage XML (extensible Markup Language) data. XML is a widely used data format for structuring & representing data, especially in the context of web services, document storage & data interchange. XML databases are optimized for handling XML data & provide features tailoured to the storage & retrieval of XML documents.

Features :

(1) Native XML Storage
XML databases store XML documents in a native format, preserving the structure & heirarchy of data, making it easier to query & manipulate

(2) Indexing
XML databases often use indexing techniques to speed

up data retrieval. These indexes can be based on elements, attribute values

(3) Update Capabilities
XML databases allow you to insert, delete, update XML documents, making them suitable for applications where data is frequently modified

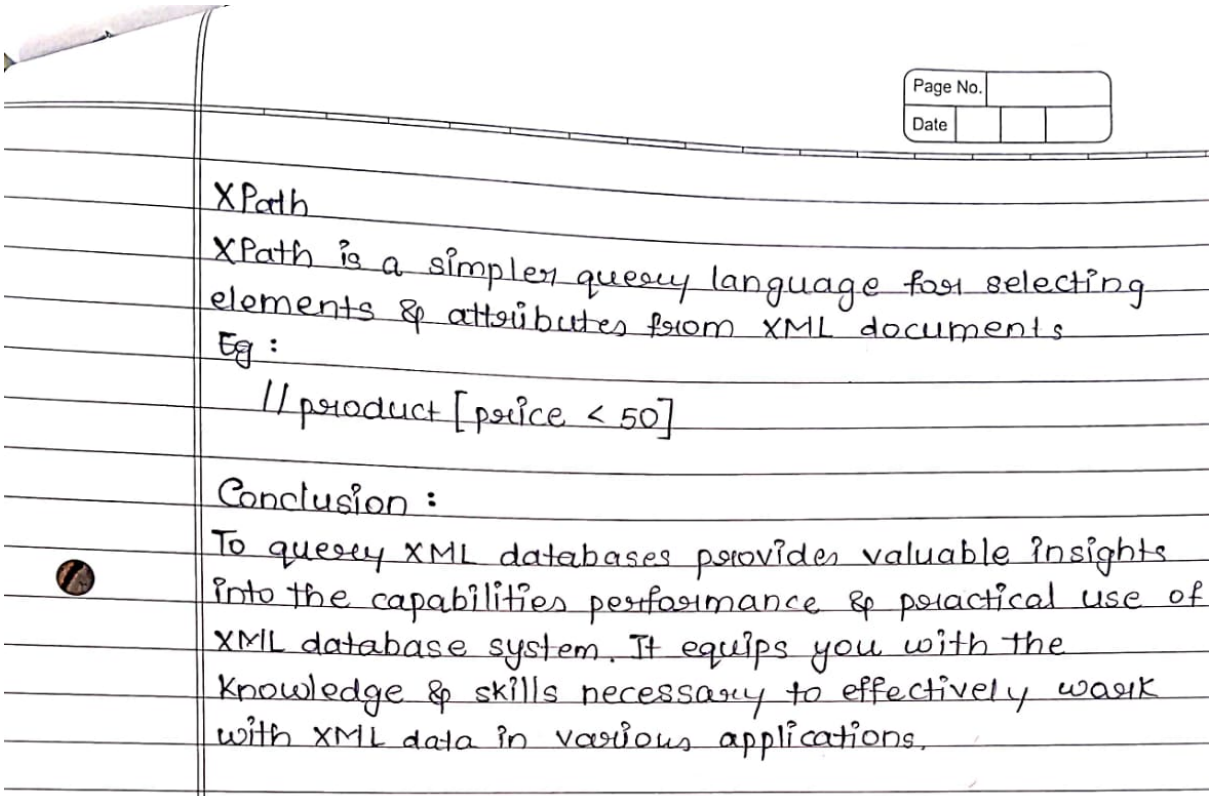
(4) Transaction Support
Many XML databases support transaction to support ensure data consistency & reliability, similar to traditional relational databases

(5) Scalability
XML database can be scaled to handled large volumes of XML data efficiently, making them suitable for applications with growing data requirements

Querying in XML databases

(1) XQuery
XQuery is a powerful query language specifically designed for querying
Eg:

```
for $book in collection("books")/bookstore/book
return $book/title
```



Code :

```
<!DOCTYPE html>
<html>
<head>
  <title>XML Catalog</title>
</head>
<body>
  <h1>Book Catalog</h1>

  <input type="text" id="titleInput" placeholder="Search by Title">
  <input type="text" id="authorInput" placeholder="Search by Author">
  <input type="text" id="genreInput" placeholder="Search by Genre">
  <input type="number" id="priceInput" placeholder="Search by Price">
  <input type="date" id="dateInput" placeholder="Search by Publish Date">
  <button onclick="searchBooks()">Search</button>

  <ul id="bookList"></ul>

  <script>
    var catalogData; // Store the XML data for searching

    // Function to load and parse the XML file
    function loadXML() {
      var xhttp = new XMLHttpRequest();
      xhttp.onreadystatechange = function() {
        if (this.readyState === 4 && this.status === 200) {
          catalogData = this.responseXML; // Store the XML data for searching
```

```
          displayCatalog(catalogData);
        }
      };
      xhttp.open("GET", "book.xml", true);
      xhttp.send();
    }

    // Function to display the catalog data
    function displayCatalog(xml) {
      var books = xml.getElementsByTagName("book");
      var list = document.getElementById("bookList");

      for (var i = 0; i < books.length; i++) {
        var book = books[i];
        var author = book.getElementsByTagName("author")[0].textContent;
        var title = book.getElementsByTagName("title")[0].textContent;
        var genre = book.getElementsByTagName("genre")[0].textContent;
        var price = parseFloat(book.getElementsByTagName("price")[0].textContent);
        var publishDate = new
Date(book.getElementsByTagName("publish_date")[0].textContent);
        var description = book.getElementsByTagName("description")[0].textContent;

        var listItem = document.createElement("li");
        listItem.innerHTML = "<strong>Title:</strong> " + title + "<br>" +
          "<strong>Author:</strong> " + author + "<br>" +
          "<strong>Genre:</strong> " + genre + "<br>" +
          "<strong>Price:</strong> $" + price.toFixed(2) + "<br>" +
          "<strong>Publish Date:</strong> " + publishDate.toDateString() + "<br>" +
          "<strong>Description:</strong> " + description + "<br>";

        list.appendChild(listItem);
      }
    }

    // Function to search books based on user input
    function searchBooks() {
      var title = document.getElementById("titleInput").value.toLowerCase();
      var author = document.getElementById("authorInput").value.toLowerCase();
      var genre = document.getElementById("genreInput").value.toLowerCase();
      var price = parseFloat(document.getElementById("priceInput").value);
      var date = new Date(document.getElementById("dateInput").value);

      var list = document.getElementById("bookList");
      list.innerHTML = ""; // Clear the previous list

      var books = catalogData.getElementsByTagName("book");

      for (var i = 0; i < books.length; i++) {
        var book = books[i];
```

```
        var bookTitle =
book.getElementsByTagName("title")[0].textContent.toLowerCase();
        var bookAuthor =
book.getElementsByTagName("author")[0].textContent.toLowerCase();
        var bookGenre =
book.getElementsByTagName("genre")[0].textContent.toLowerCase();
        var bookPrice =
parseFloat(book.getElementsByTagName("price")[0].textContent);
        var bookDate = new
Date(book.getElementsByTagName("publish_date")[0].textContent);

        if (
          (title === "" || bookTitle.includes(title)) &&
          (author === "" || bookAuthor.includes(author)) &&
          (genre === "" || bookGenre.includes(genre)) &&
          (isNaN(price) || price === bookPrice) &&
          (isNaN(date) || date.toDateString() === bookDate.toDateString())
        ) {
          var description =
book.getElementsByTagName("description")[0].textContent;

          var listItem = document.createElement("li");
          listItem.innerHTML = "<strong>Title:</strong> " + bookTitle + "<br>" +
            "<strong>Author:</strong> " + bookAuthor + "<br>" +
            "<strong>Genre:</strong> " + bookGenre + "<br>" +
            "<strong>Price:</strong> $" + bookPrice.toFixed(2) + "<br>" +
            "<strong>Publish Date:</strong> " + bookDate.toDateString() + "<br>" +
            "<strong>Description:</strong> " + description + "<br>";

          list.appendChild(listItem);
        }
      }
    }

    // Load and display the XML data when the page loads
    window.onload = function() {
      loadXML();
    };
  </script>
</body>
</html>
```

Database (sample book added here)

```
<?xml version="1.0"?>
<catalog>
   <book id="bk101">
      <author>Gambardella, Matthew</author>
      <title>XML Developer's Guide</title>
      <genre>Computer</genre>
      <price>44.95</price>
```

```
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
    with XML.</description>
  </book>
</catalog>
```

Screenshots :