Chapter 4  
Research Methodology

4.1 Introduction

Over the past decade, the expansion of internet and complex web application such as e-commerce, e-learning require data storage of different format. Since February 1998, the eXtensible Mark-up Language (XML) has become the standard medium for data representation and exchange over the Web [2]. Due to its simplicity and user friendliness, use of XML technology is increased day by day.

It is used as International Standard for document exchange. It is vendor independent and not tied up with any company or with any application. The name ‘eXtensible’ suggests that it is not restricted with fixed tag like HTML. The company or developer adds their own tag as per their need. Another important characteristic of XML is reusability, instead of creating data from scratch; XML reuses the exiting data and build the new structure from it. Another important characteristic is compatibility, it is easy to apply different style sheets to an XML document to manipulate content for different audiences or output to different documents.

Its utilization is not only restricted to document representation and manipulation but also used as functional language as well as data manipulation language. XML is capable of storing different company documents like product manual, company work process etc are created by XML. It works as functional or programming language also. XML also manipulate data and works as database management system. For example the web content like user navigation, maintenance of user profiles etc are managed by XML. These data management is nothing but the database management. Once the XML is used as database management system, it must follow the features of DBMS like query optimization, indexing, transaction management etc. Thus, a proper mechanism is needed for this management. This thesis addresses the query optimization and indexing of XML. Hence, the main objective of the thesis is determining effective storage management and index structure for XML data.
4.2 Importance of the Study:-

XML is an important component of web application. It defines the standard framework for e-business so that business communication is easy. For example, the e-commerce web site promotes their products for sell. Web site extracts the different product from different company database and preset it for sell. Their data format is different so it is very difficult to merge this information for further processing. But this is achieved through XML. XML collect these different formats and combined together and display the information into a web enable format. XML has another property of extensibility so the new records can inserted as per the need. Additional insertion cost or rebuilding cost is avoided by the XML technology. One of the important examples is EDI. The EDI is Electronic data interchange community, which was used by the large company for selling and buying their product. This service was not affordable by the small company due to high paying cost and as large company have major stock holder of it hence they have their say for changing their standards which also affect small company business. These standards are replaced by XML. XML based e-commerce is penetrate down even to small companies. Most of the people agree that XML is easy to work with more extensible

Now days, the business are need to communicate each other. Therefore the data is transfer from not only one web site to another website but also from one system to another system, from one business to each other e.g. Enterprise Application Integration. This integration application is within the company itself means accounting system is integrated with inventory system, inventory is associated with billing system. These communication is performed the XML protocol only.

XML is used not only data representation but also used for document representation like product manuals are represented using XML. It also used for metadata representation. E.g. system configuration data, the workflow procedure in the company like production cycle of the product is explain in XML. It represents serializing complex data into non-relational data. It integrates with other system.

In software application contains structure and hierarchical information. These information is very difficult to represent by any RDBMS system e.g. users preference for website. Their likes, their
navigational flow etc. cannot store by RDBMS system but XML is capable of storing this
information. As mention above workflow of any production system is unable to store by
RDBMS system but it is easily stored in XML technology.

Other important characteristic, of XML is multi-language support. The different languages
systems can communicate to each other. e.g. ‘Chines’ language website can communicate to any
‘Marathi language ‘web’ site. The translation between these two sites is performed with XML
support.

XML is also used as content management system. Content is unit of digital information. It can be
text, images, graphics, sound etc. that is likely managed in an electronic format. Content
Management is effectively the management of the content. These content re-use saves companies
money and makes authors more efficient. XML supports and promotes these sorts of processes
by being flexible and modular. XML allows you to separate content from format. The formatting
of the XML document is inside a separate style sheet. This separation allows easily maintaining
and updating formatting as needs change. It is easy to maintain a consistent style for all
documents when the content is separate from the formatting.

As internet evolved and it’s potential for e-business is cleared then W3C developed XML
language for performing the data exchange between e-businesses. There is a strong momentum
for XML to be an important “next-generation” language for E-commerce. And several
companies have been started to developed XML based solutions. All above mention
characteristics suggest the significance of XML in software.

4.3 Scope of the study:-

As the XML utilization are increased. It is used by all three types of communities-‘Document
type of Community’, ‘Data type of Community’ and ‘Functional communities’ are interested to
use XML. The ‘Document Community’ is using XML for document storage. ‘Data Community’
is using XML for data storage and database application. The functional Community is using
XML for programming purpose. As need of communities are different, the different data
retrieval or different types of queries are needed. For ‘Document Community required IR
techniques for retrieving data. Functional communities are using all programming constrains and
data communities are using data bases queries.
From 1990, onwards different query languages were developed. The different query language and their characteristics and advantages are explained into chapter2 page no28. The XQuery is languages which support all types of community. Therefore the research scope is limited to XQuery language.

This research limits its scope to ‘Data Community’. The ‘Data Community’ is interested for database functionality. The DBMS functionality is ‘Query Processing ‘, ‘Transaction Management’, ‘Database Security’ etc. This research scope limited to the query optimization techniques. In query optimization ‘indexing’ is an important component for effective query evaluation. The research scope is limited to indexing techniques. Thus the research scope is identification of effective indexing techniques for XML query execution and identification of a proper XML storage technique.

4.4 Problem Statement of the Research

Considering the above issues, the researcher defined the problem statement of the research.

Designing a flexible and efficient data storage structural for XML file and a query optimization algorithm on it. This query optimization technique includes designing of efficient index structure. This indexing structure is such that it helps in execution of the all types of query.

This new framework includes the following features:

- Assigning a unique identity to each XML node.
- The XML node identifier represents P-C, A-D, and sibling relationship. So, that is can answer all types of query like P-C, A-D queries, Partial match and Content- query by single lookup.
- This framework must avoid redundant traversal of XML tree for answering the all types of XML query.
- This index structure must reduce disk I/O cost and provide high-quality query performance result.
4.5 Objective of the Research

The objective of the research is:-

- Recognize the importance of XML in software development.
- Identification of effective storage method for XML storage and query processing.
- Designing a new index structure for XML file so that it reduced redundant traversal of XML file.
- Designing an efficient algorithm for query processing.

4.6 Statement of Hypothesis:

1. Instead of using XML in isolation, better results can be obtained by using combo model which includes RDBMS and HTML.
2. Existing model of XML does have some weak points.
3. XML-RDBMS storage is always better than XML-Native storage and XML-Navigational storage.
4. There is no difference between query processing cost of XML databases and RDBMS database.
5. XML index size is directly proportional to query processing cost and effective organization and data retrieval of XML data file.

**H₁ Instead of using XML in isolation better results can be obtained by using combo model which includes RDBMS and HTML.**

XML is an important component of today’s technology because XML has extensible, self-descriptive and application independent features. The extensibility means the data can be added or deleted from files easily. XML cannot consider backward compatibility issues so someone can add the new node into XML file; without any modification of previous data. XML is self-defining; it describes the structure and meaning of the data content that it holds. The application independence means XML can be used to exchange data among the different computer systems irrespective of the platforms on which those systems were running on. All these features suggest that XML is used in today’s e-business as well as in different e-documents easily. This
hypothesis studies the XML utilization in different application and how it is used in different e-business and XML is only sufficient for performing all business transaction as well as representation
To study this hypothesis, the case study method was followed. Different case study was selected from different company portal and identified their bottle neck problem in the business and their solution towards that problem. The study also performed on sample data collected from different software users.

**H₂ Existing model of XML does have some weak points**

The literature suggests XML utilization in the different business. It is specially used for data transformation, data communication and data representation. The subsequent list mentions the utilization of XML:-

1. The application required to store menu data or some combo box information.
2. It is used for developing the database driven websites.
3. In image gallery application, XML is used as the data file (xml) for storing the names and location of the images.
4. For shopping application, it is used to store the product details and a purchase detail of the customer as well as it is also used for performing e-payment.
5. In travel applications, XML is used as a gateway for different booking transaction.
6. The web services such as Weather services, Currency rates service etc. are using the XML language as a programming language as well as database for supplying information to process for further transaction.
7. The developers use the XML data files to generate the dynamic content by applying different Style sheets.
8. XML is also used to develop the content management systems.
9. Many companies are using XML files for writing the documents. Then program like DocBook can be used to generate the required documents.
10. Many software development frameworks are using XML files to store the configuration data for the application. For example, JPA uses persistence.xml file for configuring the JPA environment.
11. Many companies are using XML for electronic data exchanges for transferring their business records from one web site to another.
12. XML is used for transferring data information between two web sites.
13. XML is also used very easily in the application where database and flat files are difficult.

Since XML data is used for different purposes, there may be additional support required for manipulation, organization of data. This hypothesis checked the same. For the study, researchers select the different website and study how these companies store this information for representation of the data.

H₃ XML-RDBMS storage is always better than XML-Native storage and XML-Navigational storage.

XML is used by different communities: database community, functional community and document community. The document communities use XML for storing and retrieving data information. The programming communities are using XML for programming purpose. The database communities are using XML for the database.

To satisfy this requirement, XML needs to store the application data effectively and efficiently. In literature, three techniques are used for it. They are: XML-RDBMS and XML native storage and XML navigational storage.

In XML-RDBMS storage, XML data is converted into relational table in such a way that the storage maintains the P-C relationship and A-D relationship. The major advantage of this storage is it can use existing RDBMS like query optimization, indexing, data security is applicable to it. In this storage, XML queries are first converted into SQL queries, and then existing query optimization technique is used for query evaluation.

In XML-Native database storage, a specialized storage technique is derived for XML storage. It stores XML data by using different data structure like link list, stack, and multiple links lists etc. The additional data conversion cost is not required; it automatically maintains P-C, A-D and sibling relationship. All database features like normalization, query optimizations are needed to derive from the scratch. XML queries are also stored using different data structure and specialized query optimization algorithm is executed on it.

In XML navigational storage, XML data file is arrived as data stream without associated index; entire XML tree needs to traverse node by node in depth wise manner. XML queries are also traversed as a data stream. When desired match is found, the results are displayed.
This hypothesis tests the best XML storage methodology. The investigation is performed on query execution speed, index techniques and disk utilization cost. The three different products: Saxon, IBM DB2 and eXit are used for study.

**H₄ There is no difference between query processing cost of XML databases and RDBMS database**

The main goal of the query execution is retrieval of desired data with minimum time. For this purpose, proper query optimization plan is required. The optimization plan is nothing but writing the blueprint of physical and logical query execution plan for effective retrievals of data from the query and query optimizer selects the most efficient plan from it. The query execution plan calculates the query cost. The query costing parameters are ‘Data Organization’, ‘Query Execution time’, ‘Index size’. This hypothesis checks the cost of query processing in RDBMS system as well as in XML system. It also studies the different impact of different factors on query execution.

The experiment is performed on IBM DB2 and SAXON for XML query processing. In IBM DB2, the data is stored into RDBMS formant. In SAXON, the XML file is text file and java Saxon package executed for query execution.

**H₅ XML index size is directly proportional to data organization of XML file and query processing cost.**

In query execution, indexing plays an important role. Indexing is nothing but an organization of data in such a way that the searching value can be retrieved data effectively with minimum time. These indexes are created on some data filed. These data filed is nothing but search key value in the query.

XML is a hierarchical order, recursive nature of data. The data searching and data retrieval is very difficult. These operations can perform effectively, if the data is organized in effective manner, this hypothesis tests how the XML data is organized and studies the different XML-indexing techniques on it. This hypothesis also checks the impact of index size on XML storage and query execution.

To test this hypothesis, the different index types are selected (node index, path index and combination of node index + path index). Researchers perform the experiment with different size of XML files i.e. XML file contains 10 records, 100 records and 1000 records and same index structure. It measures performance i.e. query execution time.
4.7 Research Methodology :-

This research study is limited to identification and designing a framework for the best XML storage and data retrieval technique. It utilizes both primary and secondary data. The primary data is collected through questionnaire, experiment, case study analysis and content analysis. The scope of research is limited; the survey is undertaken by obtaining a purposive and quota sample. The description of the research methodology required for the process of obtaining a sample as well as the nature and size of sample should be adequately explained. Probability quota, stratified sampling technique were used. The selection of respondents was involved based on the important characteristics under study such as XML known users, software developers and software tester etc.

4.7.1 Secondary Data:-
XML satisfied the need of all three communities: document community, data community and functional community. It is used for data representation, data community is used XML for data storage and query evaluation. For functional community it is used as programming language. Hence all community has interest in XML development. All these communities had performed extensive research in their own area. One of the common problems between them is designing of a query optimization techniques for data retrieval from XML database. Therefore, this thesis is focus on indexing and query processing techniques for data retrieval from XML database. The secondary data was collected from different website, different articles, periodical and the different research articles. The secondary data is collected analysis.

4.7.2 Primary Data :-
This research is for designing new indexing and query processing technique for XML. For our study, the researchers selected following methods for data collection:-

(1) Questionnaire: - XML is an important constituent in web software development. This method was used for collecting the opinion of different software experts.
(2) Experimental Research: - This method was adopted for testing the XML storage and query evolution techniques.
(3) Case Study Method: - This is one of the important methods for research. The data was collected from the different company cases. The analysis was done on identification of the problem, the XML solution towards that problem.

(4) Content Analysis: - It consists of analyzing the contents of documentary material. In this research, XML utilization in different business was analyzed by using this method.

The primary data collected by the researcher is explained in the following manner.

4.7.3 Questionnaire: -

The method was used for collection of different data from different software users. The main focus from this method was analyzing the XML utilization from the different user’s perspective. Small questionnaire was set and distributed to software user. The samples were selected as Industry users and non industry users. Non industry users are computer teacher. From the industry the experts were classed as ‘Software developer’, ‘Software tester’, ‘other category ‘(they may be in maintenance of product or on bench people). The sample size is as follows:-

<table>
<thead>
<tr>
<th>Software Role</th>
<th>No of sample Collected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Computer Teacher</td>
<td>40</td>
</tr>
<tr>
<td>2 Software Development</td>
<td>470</td>
</tr>
<tr>
<td>3 Software Testing</td>
<td>580</td>
</tr>
<tr>
<td>4 Support Team</td>
<td>112</td>
</tr>
<tr>
<td>5 Other in software Company</td>
<td>136</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1338</strong></td>
</tr>
</tbody>
</table>
4.7.4 Experimental Research:

The laboratory research was conducted for studying XML storage and indexing method.

(1) Software used for experiment: Table 4.2 lists the software used for experiment.

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Software used</th>
<th>XML type Storage</th>
<th>Index Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM DB2</td>
<td>Native Storage- Coarse-Grained (special data type .xml is used for storing XML file)</td>
<td>Path Index</td>
</tr>
<tr>
<td>2</td>
<td>IBM DB2</td>
<td>Fine-Grained (RDBMS Storage )</td>
<td>B+ index is used</td>
</tr>
<tr>
<td>3</td>
<td>SAXON</td>
<td>Native Storage – Navigation approach for storage</td>
<td>Not possible to create Index</td>
</tr>
</tbody>
</table>

(2) Data Set for experiment: XML data can store into three types of file: Document-centric, Data-centric, Mix-mode (combination of Data-centric and Document-centric application). In Document-centric XML documents store text information e.g. user’s manuals, static web pages, and marketing flyers or brochures. They are characterized by loose, irregular structure. Their structural sequence is crucial. For storing this type of data, content management system tools which are used commercially. The available tools are SyCOMAX, Content@ and Frontier, etc. Data-centric documents are used for data exchange and transport medium. They are usually highly structured and marked up with XML tags. Data-oriented paradigm has focused only on data stored information e.g. web service return; current stock price, stock information, Railway reservation etc. Third XML storage is ‘Mix-mode’ which is a combination of data-centric and document-centric storage. Mix-mode data is preferred to store into native database [3]. For the experiment, the following types of files were considered: -
Table 4.3: Input Data Files for Experiment

<table>
<thead>
<tr>
<th>File Type</th>
<th>Input Data information</th>
<th>Depth of File</th>
<th>Breadth of File</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Set-I</strong></td>
<td>Data Centric File</td>
<td>Set1</td>
<td>Constant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set2</td>
<td>Varying</td>
</tr>
<tr>
<td><strong>Data Set-II</strong></td>
<td>Mix Mode File</td>
<td>Set3</td>
<td>Constant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set4</td>
<td>Varying</td>
</tr>
</tbody>
</table>

1. Data Centric File: - Two files are considered
   a. CDINFO.XML: - Depth of the file is four and six files with 10 nodes, 100 nodes, 1000 nodes, 10,000 nodes, 50,000 nodes and 100000 nodes.
   b. WHETHER.XML: - Depth of the file is ten and six files with 10 nodes, 100 nodes, 1000 nodes, 10,000 nodes, 50,000 nodes and 100000 nodes.

2. Mix-mode File: - These files are combination of data centric as well as document centric.
   a. Book.XML: - Depth of the file is four and six files with 10 nodes, 100 nodes, 1000 nodes, 10,000 nodes, 50,000 nodes and 100000 nodes.
   b. Chap.XML: - Depth of the file is ten and six files with 10 nodes, 100 nodes, 1000 nodes, 10,000 nodes, 50,000 nodes and 100000 nodes.

The detail Storage Structure of above files is listed into following table 4.4 and detail file structure is explained into annexure –II.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of node at 1st Level (Assume root level no 0)</th>
<th>Max fan-out (maximum node at a particular level)</th>
<th>Depth of the file</th>
<th>Total number of nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 1.1 : Data Centric File : CD.XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET 1.1.1</td>
<td>10 - nodes</td>
<td>60</td>
<td>4</td>
<td>197</td>
</tr>
<tr>
<td>SET 1.1.2</td>
<td>100 - nodes</td>
<td>600</td>
<td>4</td>
<td>703</td>
</tr>
<tr>
<td>SET 1.1.3</td>
<td>1000 - nodes</td>
<td>6000</td>
<td>4</td>
<td>6301</td>
</tr>
<tr>
<td>SET 1.1.4</td>
<td>10000 - nodes</td>
<td>59952</td>
<td>4</td>
<td>69944</td>
</tr>
<tr>
<td>SET 1.1.5</td>
<td>50000 - nodes</td>
<td>300000</td>
<td>4</td>
<td>350140</td>
</tr>
<tr>
<td>SET 1.1.6</td>
<td>100000 – nodes</td>
<td>599994</td>
<td>4</td>
<td>699994</td>
</tr>
<tr>
<td>SET 1.2 Data Centric File : Whether.XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET 1.2.1</td>
<td>10 - nodes</td>
<td>150</td>
<td>12</td>
<td>483</td>
</tr>
<tr>
<td>SET 1.2.2</td>
<td>100 - nodes</td>
<td>1575</td>
<td>12</td>
<td>5041</td>
</tr>
<tr>
<td>SET 1.2.3</td>
<td>1000 - nodes</td>
<td>15075</td>
<td>12</td>
<td>48241</td>
</tr>
<tr>
<td>SET 1.2.4</td>
<td>10000 – nodes</td>
<td>150000</td>
<td>12</td>
<td>471006</td>
</tr>
<tr>
<td>SET 1.2.5</td>
<td>50000 - nodes</td>
<td>749880</td>
<td>12</td>
<td>2349530</td>
</tr>
<tr>
<td>SET 1.2.6</td>
<td>100000 – nodes</td>
<td>1500045</td>
<td>12</td>
<td>5000145</td>
</tr>
<tr>
<td>SET 2.1 Mix mode File : Chp.xml</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET 2.1.1</td>
<td>10 - nodes</td>
<td>40</td>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td>SET 2.1.2</td>
<td>100 - nodes</td>
<td>402</td>
<td>5</td>
<td>699</td>
</tr>
<tr>
<td>SET 3.1.3</td>
<td>1000 - nodes</td>
<td>3920</td>
<td>5</td>
<td>6861</td>
</tr>
<tr>
<td>SET 3.1.4</td>
<td>10000 - nodes</td>
<td>39200</td>
<td>5</td>
<td>68601</td>
</tr>
<tr>
<td>SET 3.1.5</td>
<td>50000 - nodes</td>
<td>196000</td>
<td>5</td>
<td>343001</td>
</tr>
<tr>
<td>SET 3.1.6</td>
<td>100000 - nodes</td>
<td>392000</td>
<td>5</td>
<td>686001</td>
</tr>
<tr>
<td>SET 2.1 Mix mode File : MXDB.xml</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET 2.2.1</td>
<td>10 - nodes</td>
<td>40</td>
<td>10</td>
<td>181</td>
</tr>
</tbody>
</table>
(3) Query Set for Experiment:

The XML queries are classified as: containment queries, order queries and node queries or content queries [19]. The query syntax is classified into two types: (1) Tree Structure queries (2) Containment Query. (3) Twigs Queries (4) Containment queries.

Table 4.5 Types of XML Query

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Query Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Tree Traversal</strong></td>
</tr>
<tr>
<td></td>
<td>Following</td>
</tr>
<tr>
<td></td>
<td>Following – sibling</td>
</tr>
<tr>
<td></td>
<td>Preceding</td>
</tr>
<tr>
<td></td>
<td>Preceding - sibling</td>
</tr>
<tr>
<td>2</td>
<td><strong>Containment Query</strong></td>
</tr>
<tr>
<td></td>
<td>Child axis query</td>
</tr>
<tr>
<td></td>
<td>Self axis query</td>
</tr>
<tr>
<td></td>
<td>Descendent axis query</td>
</tr>
<tr>
<td>3</td>
<td><strong>Twig</strong></td>
</tr>
<tr>
<td></td>
<td>FLOWR- Clause - return small XML tree</td>
</tr>
<tr>
<td>4</td>
<td><strong>Content matching query</strong></td>
</tr>
<tr>
<td></td>
<td>FLOWR- Clause - match the content of the query</td>
</tr>
</tbody>
</table>

The following eighteen queries are selected for the experiments which satisfy all above mentioned requirements. The Syntax of the queries is listed into Table 4.6 :-
Table 4.6  Query Set for Experiment

| Q1   | select a complete tree | for $c$ in doc("CD.XML")/CATALOG/CD return $c$ |
| Q2   | Select a particular tag of tree | for $c1$ in doc("CD.XML")/CATALOG/CD return <item>{$c1/TITLE}$ |
| Q3   | Following | for $c1$ in doc("CD.XML")/CATALOG/CD[5] return {$c1/following::*} |
| Q4   | Following-sibling Axis | for $c1$ in doc("CD.XML")/CATALOG/CD[5] return {$c1/following-sibling::*} |
| Q5   | Following-sibling Axis | for $c1$ in doc("CD.XML")/CATALOG/CD[1] return {$c1/following-sibling::*} |
| Q8   | Preceding-sibling Axis | for $c1$ in doc("CD.XML")/CATALOG/CD[1] return {$c1/preceding-sibling::*} |
| Q9   | Self-axis | for $c1$ in doc("CD.XML")/CATALOG/CD[5] return {$c1/self::node()} |
| Q12  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR=1985 return $c1 |
| Q13  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR>1985 return $c1 |
| Q14  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR<1985 return $c1 |
| Q15  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR<1985 return $c1/TITLE |
| Q16  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR=1985 return $c1/TITLE |
| Q17  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR=1985 return $c1/YEAR |
| Q18  | FLOWR | for $c1$ in doc("CD.XML")/CATALOG/CD where $c1/YEAR<1985 return $c1/YEAR |
(4) **Software and Hardware Requirement:**

- Hardware Configuration: Intel Pentium (R) Dual Core CPU with 2.69GHz CPU, 2.79GB RAM and 40 GB hard disk and Windows Operating System.
- Software: For relational database implementation researchers use IBM DB2 Pure XML 9.0, Java for PESS algorithm and c++ language for XML node_id creation.
- Parameter Set UP:
  - Navigation Approach: SAXON
  - JDK1.6.0
  - Heap Size
  - Saxon9he
  - For DB2 Pure XML researchers set the following parameters:
    - Page Size : 4 KB
    - Buffer Pooling
    - DB2 Pure XML: The indexes are set on XML files ‘path encoding’ sequence and secondary index on node_id values.
  - Researchers select JDK 1.6.0 and visual C++ for the experiment. Additional parameter setting is not required for it. Only researchers change Java Heap memory for reading XML files.

(5) **Measurement Parameter:**

The Measurement parameters are:

1) **XML Storage Measurement:** XML storage is measure on maintains of the structural relationships between XML nodes. XML data is represented into tree structure format, therefore the maintenance of XML structure and actual stored data information are two important components of query execution. Under this section the researcher measure the two parameters (1) XML storage structure. (2) Data retrieval technique.

2) **XML query measurement:** As per the [6] [19] [7] proposal’s guidelines the following parameters are listed for query evaluation:

- Query evaluation speed: This criterion refers to the time required to evaluate a query in comparison to different XML storage and different query traversal.
- Disk traffic: It refers to the amount of data that an index requires to evaluate a query, i.e. the size of pages which it fetches from disk.
- Disk space: The amount of disk space required for storing the XML data.
- Scalability: The size of documents stored in databases constantly grows and the costs for storage space increases. Index structures therefore should not only to be able to handle large amount of data, but also to scale up the small documents.
- Index updates: Refers to the time which it takes to update an index when updating a document. Thereby, updates include insertion, deletions and modifications of documents.

(6) **Algorithms for Measurement:**

a. **Measurement variables for IBM DB2:**

1) **Query Evaluation Speed:** Researchers have written following program for query evaluation time measurement. For each query, it is executed 100 times and takes the average of response time.

<table>
<thead>
<tr>
<th>Algorithm 4.1: Query Measurement Time (IBM DB2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Load Database driver</td>
</tr>
<tr>
<td>2. Establish the Connection</td>
</tr>
<tr>
<td>3. Create Query statement</td>
</tr>
<tr>
<td>4. Read the start time of the query</td>
</tr>
<tr>
<td>5. Read the end time of the query</td>
</tr>
<tr>
<td>6. Repeat the steps 4 and step 5 ten times</td>
</tr>
<tr>
<td>7. Take the average of query execution time</td>
</tr>
<tr>
<td>8. Repeat the process for each query.</td>
</tr>
</tbody>
</table>
Psudocode 4.1 :- Query Execution Time Measurement

class cdq1
{
    public static void main(String argv[])
    {
        // load driver
        // create connection
        // create statement object
        // create query statement
        for(int k = 0; k < n; k++)
        {
            t1=(new java.sql.Timestamp(System.currentTimeMillis()));
            String query="XQUERY for $y in db2-
            fn:xmlcolumn('CD.CDINFO')/CATALOG/CD
            where $y/YEAR =1985 return
            $y/YEAR";
            ResultSet rs = stmt.executeQuery(query);
            // retrieve and display the result from the xquery
            while (rs.next())
            {
                String data=rs.getString(1);
                // Close the result set
                rs.close();
                // Close the statement object
                stmt.close();
                t2=(new java.sql.Timestamp(System.currentTimeMillis()));
                total[k]  = t2 - t1;
            }
        }
        // end of for statement
        System.out.println(total(k) /k/);
    }
    // end of main
} //end of class cdq1
2) **Disk Space Measurement** :-
- Disk space measurement in Navigation Approach :
  - Size of text file
  - Memory required for constructing a node structure.
- Disk space measurement in RDBMS Approach
  - Read Table space
  - Read Page Size
  - Query on Estimate Size of database provide statistic of table
  - Read database size as well as index size
- Following formula is used for disk space measurement :-
  \[
  \text{Disk size of the file} = (\text{total number of rows} \times \text{average row length}) \times (\text{total number of Pages} \times \text{Page Size})
  \]

3) **Disk traffic Measurement** :-
The query execution plan provides the ‘cardinality’, ‘selectivity’, and ‘total buffer pool pages’ used in query evaluation, CPU cost, I/O instruction cost. The disk traffic is nothing but total number of buffer pages and the I/O cost required for query execution for the particular cardinality, and selectivity with the specific CPU cost.

4) **Scalability Measurement**:-
For the experiment we used total five file with growing database i.e. from 10 records till 100000 records. Index is also defined for this entire file. Hence the parameter check index structure handles large amount of data.

5) **Index updates**:-
The new record is added, deleted or modified into the file, the index structure is capable to perform all these changes into index file.
b. Measurement Variables for SAXON

1. SAXON Query Execution Speed: - The algorithm is same as mentioned in psudocode 4.2. The time is measured by using java time library

   **Psudocode 4.2 :- Query Execution Time Measurement on SAXON**
   ```java
   class cdq1
   {
   public static void main(String argv[])
   {
   for (int k = 0; k < 10 ; k++)
   {
   t1=(new java.sql.Timestamp(System.currentTimeMillis()));
   java net.sf.saxon.Query [options] query filename
   t2=(new java.sql.Timestamp(System.currentTimeMillis()));
   total[k] = t2 - t1;
   }//end of for statement
   system.out.println(total(k) /k/);
   } // end of main
   }// end of class cdq1
   ```

2. SAXON : Disk Traffic Measurement
   o Navigation Approach :SAXON-
     o Saxon is not a database product. It cannot afford luxury of data storage and memory residential indexing.
     o XML file is read by SAX parser and Nodes are created on fly.
     o Total number of node reading is depending upon java heap size
     o I/O cost or Disk traffic depends upon heap. e.g. If heap memory is 25 bytes and file size 100 bytes file is sliced into four and I/O cost is four.

3. SAXON: Disk space measurement
Saxon is not the database product hence the disk space is calculated by amount of data –space required for storing XML file i.e. text file. SAXON is not database product hence remaining two parameters are not needed for testing it.
4.8 Case Study Method:-

Managing data into different business is a big challenge in today’s e-business. The business data interact to the different organization not only for information exchange but also reuse of the existing data in the organization in different format. In the data information exchange, the data incompatibility problem is arising. This incompatibility problem was solved by XML. It is a structure document, its aim is to separate presentation, structure and meaning from actual document. Therefore the documents can easily understand. It is easily interchange between businesses. It can reuse in multiple situation instead of creating it from scratch.

To test these characteristics researchers study the different cases of different company. These company adopted XML techniques in their organization. Researcher studies it with reference to following points - the existing business problems and their XML solution. Why companies need to use the XML technology? Study the XML utilization for different purpose. Once the data is shifted then does sold technology is needed.

4.9 Content Analysis:-

Business has urged to communicate each other not only for email, fax but also for the data entry, data interpretation and electronic transactions. Previously, EDI is used for it. But EDI depends on proprietary networks and X12 document descriptions. It is third party proprietary. Large company feels free to modify it as they pleased. This EDI is replacing by XML. An XML document can carry the description of its contents in document header section. The receiving system, read the header and handles the data, it is not depends upon the third party. Different organizations are involved in writing the ‘vocabulary ‘and designing ‘framework’ for standardization. Different web services have been designed using XML data. For the study, the following business framework was selected:-

- E-commerce :- XML used for e-transaction, electronic data exchange,
- Financial: - e.g. Financial report creation, electronic exchange of securities transactions, used for representation of structured financial products.
- Government :- Tax XML includes a vocabulary of terms, a repository of artifacts including XML templates, documents exchanged for tax compliance
- Astronomy e.g. astronomical data, such as images, spectra, tables, and sky atlases.
- Food e.g. Meat & Poultry Data Standards Organization
- Research Lab: XML data for representation of research content
- Legal e.g. used for Electronic Court Filing, e-Contracts, e-Notary

The XML Business frameworks with examples are listed in Table 4.7:

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Business Domain</th>
<th>Examples of XML utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E-commerce</td>
<td>Online Shopping</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Electronic Payments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online Auctions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internet Banking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Online Ticket Booking</td>
</tr>
<tr>
<td>2</td>
<td>Financial</td>
<td>Financial company data representation</td>
</tr>
<tr>
<td>3</td>
<td>Government</td>
<td>Government Policy data representation</td>
</tr>
<tr>
<td>4</td>
<td>Legal</td>
<td>XML is used for representation of legal matter</td>
</tr>
<tr>
<td>5</td>
<td>Research</td>
<td>Utilization of XML in research, like Space science (NASA), Basic science document representation (Math, Physics, Chemistry etc)</td>
</tr>
<tr>
<td>6</td>
<td>Computer science</td>
<td>XML data utilization in Computer Science</td>
</tr>
<tr>
<td>7</td>
<td>Publication</td>
<td>XML utilization in Publication</td>
</tr>
<tr>
<td>8</td>
<td>Food and Agricultural</td>
<td>XML utilization in food Industry and agricultural domain</td>
</tr>
<tr>
<td>9</td>
<td>Entertainment</td>
<td>XML used in entertainment</td>
</tr>
<tr>
<td>10</td>
<td>Engineering</td>
<td>Utilization of XML in Engineering</td>
</tr>
<tr>
<td>11</td>
<td>Pharmacy</td>
<td>Pharmacy company</td>
</tr>
<tr>
<td>12</td>
<td>Health care</td>
<td>Utilization of XML in healthcare industry</td>
</tr>
<tr>
<td>13</td>
<td>Other</td>
<td>Utilization of XML in space, Artificial Intelligence, Music etc</td>
</tr>
</tbody>
</table>
4.9.1 Hypothesis testing techniques :-

The researcher has collected primary data in the field work and through the laboratory experiment. The said data is properly analyzed with statistical techniques. In addition to this, necessary graph and charts have also been prepared to support the analysis of the data wherever necessary.

4.10 Summary and Conclusion

XML utilization is increased day by day and now it becomes the standard media for data exchange on the internet. One of the dimensions of utilization it is a database product. Hence effective data storage mechanism and indexing strategy is required. Due to XML characteristics it is retrench challenge for designing perfect indexing schema for effective storage.

This chapter explores the research objective and hypothesis of the research. It also discussed the prerequisites of the experiment. The input, Hardware, Software, Platform used for experiment as well as software used for experiment.
References:-

1. Abraham Silberschatz, Henry Korth, S. Sudarshan "Database System Concept” McGraw-Hill Education 27-Jan-2010

2. Barbara Catania and Anna Maddalena, Athena Vakali “XML Document Indexes: A Classification SEPTEMBER • OCTOBER 2005 Published by the IEEE Computer Society 1089-7801/05 © 2005 IEEE INTERNET COMPUTING


14. M Al-Nadawi, Dr Siobhan North, Dr Barry Eaglestone Indexing XML Databases: “Classifications, Problems Identification and a New Approach” , The University of
Sheffield Department of Computer Science technical report 15th November, 2007

15 Mag.a Katharina "Flexible and Selective Indexing in XML Databases" , Johannes Kepler Universität Linz, Altenberger Straße 69, 4040 Linz, Austria ,August 2008

16 Matthias Nicola , Bert van der Linden "Native XML Support in DB2 Universal Database " Proceedings of the 31st VLDB Conference , Trondheim, Norway, 2005


18 Michael Kay "XSLT and XPath Optimization " "http:// saxon.sf.net"


27 Xiaojie Yuan Xin Wang Chenying Wang Efficient XPath Evaluation Using a Structural Summary Index, 978-0-7695-3336-0/08 © 2008 IEEE DOI