XML and DATABASES
Introduction

1.) Context.

2.) Databases.
   2.1) Definition.
   2.2) The different models.

3.) XML (Extensible Markup Language) and relational databases.
   3.1) What is XML?.
   3.2) Storage of XML documents in a relational database.
   3.3) XML and query language.
   3.4) Publication of XML documents in a relational database.

4.) XML and federated databases.

Conclusion

Bibliography

Glossary
**Introduction**

The explosion of internet and the progress of technologies give some new approaches on databases used in information systems. It has some consequences on the management of information which becomes more complicated. Firms are some good consumers of big and heterogeneous information. So in order to answer their needs, some new standards such as XML: extensible markup language are trying to make and to represent information.
1.) **Context.**

Today web applications become more and more sophisticated and need to access to a full different type of information. These applications are the diffusion of information such as publicity, the exchange of information such as news, the search of information which uses some search web engine and finally the web business such as purse activities or transactional commercial activities which develop and increase every day. Now a lot of firm open their system of information to the internet, and so some new kind of architecture appear. The development of internet and firms’ technologies show that the web applications need to couple internet technologies and databases. Today clients access to the databases through the web and so it needs a special architecture in order to answer to their requests. In parallel, Web sites are developing every day. The web sites become very numerous and very heterogeneous. Their contents are very complex for example we can find several HTML files with different links and it has became very difficult to modify it when it is necessary that’s why database is a good solution to solve this problem. It gives a method to represent information and organize a logical hierarchy between information.

2.) **Databases.**

2.1) **Definition.**

A database is a collection of files (entity) which are binding with logical links and organizing in order to answer efficiently to a big variety of questions.

2.2) **The different models.**

The first model is the relational model. It must centralize information and insure that data and data processing are independent. Moreover, this model insures the links between data, and the rules which are:
- The rules of integrity and coherence.
- The rules of security and confidentiality.

This model must represent the universe in table of two dimensions. The table is calling relational table. All the model is creating on a schema for example on figure 3, it represents an object “CLASS” which is identified by its primary key “NumClass”, “ROOM” which is identified by its primary key “NumRoom” and “SUBJECT” which is identified by “NumSubject”. This key is unique and identifies each object “CLASS”, “ROOM” and “SUBJECT”. The information will be obtained by asking some requests on this table by using a query language SQL: Structured Query Language.
Figure 3: Example of a relational schema

For example the relational table CLASS is:

<table>
<thead>
<tr>
<th>NumClass</th>
<th>Nbstudent</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
</tr>
</tbody>
</table>

The advantages of this model are that the query language is based on a solid theory algebra rules and it is simple to build. But the inconvenient of this model is that the schema is fixed and it is difficult to modify it.

So the second model is a model of files. It has a simple structure and so it is easier to modify it than the first model. This model consists to collect files and it is organizing as the system of files of an operating system such as we have on a computer. In this model, the big advantage is the simplicity for managing files. But there are no information on the structure and no schema so the consequences are that information can be redundant. But the real problem is that there is no query language for this model.

The third model is the semi structured data which must answer to the facts of that:

- The data can not be compatible with a schema.
- The structure of the data evolves and needs to be updated frequently.
- The data can be to structure slightly.

It is a compromise between the relational model and the model of files. So there is a supple schema and a query language. This model appeared in order to manage heterogeneous and structured data.
3.) **XML (Extensible Markup Language) and relational databases.**

3.1) **what is XML?**

XML was born of the SGML (standard generalized markup language) from the W3C consortium. W3C are a group of 400 firms Oracle, IBM….and laboratories research as INRIA (Europe), MIT(USA). W3C must define a model to facilitate the exchange of data on internet. XML was created in order to uniform document structure. XML is independent of the internet browser and it gives the possibility to represent the same document in some different formats such as HTML or pdf, ps…. .

XML defines a standard model of data exchange. As HTML, XML has got some tags which represent the structure’s definition. Figure 4 shows an extract of XML code.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<!DOCTYPE Guide SYSTEM "travel.dtd">
<Guide version="2.0">
  <Restaurant>
    <Name> Aubergeade </Name>
    <Street> Des paris </Street>
    <Phone> 0148152256 </Phone>
    <Manager> Dupuis </Manager>
  </Restaurant>
  <Restaurant>
    <Name> La Licorne </Name>
    <Street> Des Moines </Street>
    <Phone> 0148253278 </Phone>
    <Manager> Dupuis </Manager>
  </Restaurant>
  <Bar>
    <Name> Rose and Crown </Name>
  </Bar>
</Guide>
```

Figure 4: An extract of XML code which describes bars and restaurants.

The second line gives the access of the DTD (document type definition). The DTD defines the tags for example the tag `<Name>` and the schema of the document. These tags must be in the right order. First of all, the document must begin with a first tag and finish with an end tag. The tags frame the elements. Each element is linking with a simple or complex attribute. They can be integrating in other elements. A document is simply a continuation of elements which are built into other elements.
A XML document is building as a tree graph:

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<cinema>
  <name>epee de bois</name>
  <address>100 rue Moufflard</address>
  <subway>Censier-Daubenton</subway>
</cinema>
```

Figure 5: a view of XML document

3.2) **Storage of XML documents in a relational database.**

XML documents are translating to graphs and relational table. This technique of storage called “mapping” is supported by a large majority of commercial relational database. XML documents are represented as a labeled graph. Each rainbow represents a XML tag and the knot leafs are the data elements (figure 6).

```xml
<restaurant id=1>
  <name>Aubergeade</name>
  <manager>Dupont</manager>
  <address>
    <number>30</number>
    <street>Des Vignes</street>
  </address>
  <drive tid=7/>
</restaurant>
<car id=7>
  <mark>BMW</mark>
  <color>blue</color>
</car>
```

Figure 6: An example of a storage XML document in a relational database.(source from eXMLmedia)
3.3) **XML and query language.**

Several language of requests are:

- **XPATH 2.0** works on a XML document. It gives the possibility to extract nodes and puts a reference on each node of the tree.
- **XSLT 2.0** which can translate a XML document to XML ,HTML or text.
- **XQUERY 1.0** comes from XPATH 2.0. It is using in order to access to the database and it is a real query language.

3.4) **Publication of XML documents in a relational database.**

The three languages are used in the figure 7. It shows the process to have access to the database.

![Diagram](image)

**Figure 7:** functionnality

Today XML is using for publication and exchange data. In parallel, Database is using for transaction and to update data. In a database all the data are structured and they have got a type but in XML there are simple structure, no type and a lot of text. So the advantage is that XML can publish a part of the data from a database and can be used for generating different document format.
4) **XML and federated databases.**

XML can be used in a federated database. A federated database is an heterogeneous database which is made of different data such as text files, HTML documents XML…….The objective is to give an integrated view to the users of the different data from the enterprise. So the idea is that all the requests from the user are XML requests. The XML requests will be translated in the right compatible source language.

![Figure 8: An example of a component which were made by the eXMLmedia Enterprise.(source from eXMLmedia).](image)

The XML-QL requests from the user go to the mediator which traduces them in a compatible language with the source destination.

**Conclusion:**

XML and the semi structured model make the integration between web and database easier. Today the problem is that a lot of documents from firms need to be translated into XML files but we must find what kind of important information in the documents should appear. It is the domain of the data mining which represents the future generation of intelligence tools. Data mining consists to extract a part of information from a big source of information and makes it exploitable. Also XML will have a good future.
Bibliography:

Online resources

http://www.e-xmlmedia.fr/site_francais/documentation_livres_blancs.htm
Georges, Gardarin. Les datawebhouses arrivent

BSTAT. Generalites sur les bases de donnees.

www.blois.univ-tours.fr/~mirian/mi-BD3.html

cedric.cnam.fr/vertigo/Cours/ Valeur-C/bamain-valc-4x.pdf
Bern Amann et Philippe Rigaux. Donnees semi structurkes et XML

lithwww.epfl.ch/teaching/docmul/15-bd/bd-slides.pdf
Dr Yanne Rikik. XML et les bases de donnees

www.httr.ups-tlse.fr/pedagogie/cours/bd/bdrel.pdf
M.Boughanem. Bases de donnees relationnelle

Text resources

B.Defude. XML et Bases de donnees
Glossary:

DTD: document type definition
HTML: HyperText Markup Language
SGML: Standard generalized markup language.
XML: Extensible Markup Language
XML-QL: Extensible Markup Language-Query Language
XQuery: Extensible Query
XSLT: Extensible Stylesheet Language transformation