Design and implementation of an XML to relational database translator

Yasmin Muwafaq Kassim
Assistant Lecturer
College of Engineering
Al-Nahrain University

Abstract

Recently, XML (eXtensible Markup Language) is the common format for representing , exchanging, storing, and accessing data presents many new challenges to database systems. Since the majority of everyday data is still stored and maintained in relational database systems, we expect that the needs to convert data format between XML and relational models will grow substantially. So this research presents a translating algorithm which programmed by visual basic to convert data from database to XML(for one to one relation).

The program request the database as input and produce XML as output, the translator will convert database name to root node, central table to central element, all fields name are converted to child elements to the central element, all values of fields in each record are converted to values of elements in each iteration and finally the primary and foreign keys are converted to attributes to central element. This translator has implemented on many databases but here we mention only two (Email registration and class registration) with there results (XML file).

XML database open new query and processing capabilities. Duplication of content has been eliminated, leading to better consistency of information and text searching has been enhanced, leading to more robust information retrieval.

Keywords: Relational Data Base, XML.

1. Introduction

An XML document is a repository for data. A database is also a repository for data. Therefore, it makes perfect sense to state that the two are the same in some respects. Data base consists of tables.

A table is the structure applied to repetitions of some data item. For example, in order to store customers in a database, you need to create a table for those customers. Those customers have names and addresses.

A customer table consists of a field for the customer name and a field for the address of the customer. A record is the name and address of each customer. Each record is a repetition of the structure of the table (the name and address fields). If there is another table for telephone numbers which contains customer telephones, there must be a relation between the first table and the second, this relation is done by setting a primary key for the first table and a foreign key for the second; this structure results a RDB which consists of two tables related to each other by one to one relation [1,2].

The abbreviation "XML" refers to eXtensible Markup Language, which means that XML is extensible or changeable. And also allows generation of web pages on the fly. XML allows storage of changeable data into web pages that can be altered at any time besides runtime. XML pages can also be tailored in look, feel, and content, and they can be tailored to any specific user looking at a web page at any point in time [3,4].

The research displays the lines below as an example to XML document:

```
<customers>
<customer>
<name>Ahmed</name>
<address>Mansur</address>
</customer>
<customer>
<name>Mohammed</name>
<address>Babil</address>
</customer>
< /customers>
```

When there is another table(telephone) related to customer table; the preceding XML format must be altered to contain telephone field and the primary key will be an attribute for the customer element.

After displaying XML and RDBMS, there is a need to convert from one to the other , the research presents an algorithm programmed with visual basic to convert from RDBMS (one to one relation) to XML database and this will be presented in the next chapter.

1.1 Related Work

1. Converting Legacy Relational Database into XML Database through Reverse Engineering [5]: In this paper the research first applying the reverse engineering approach to extract the ER (Extended Entity Relationship) model from a legacy relational database, then convert the ER to XML Schema. The proposed approach is capable of reflecting the relational schema flexibility into XML schema by considering the mapping of binary and nary relationships.
2. A Cost-Based Approach for Converting Relational Schemas to XML [6]: the research propose a cost measure based on the space efficiency of the XML encoding as a metric for measuring the XML schema's desirability. Using this cost metric, the researcher define an algorithm that returns the optimal XML encoding exhibiting no redundancy.
3. Converting Relational Database into XML Document[7]: This paper provides a methodology of translating the conceptual schema of a relational database into XML schema through EER (extended entity relationship) model. Physical data are then translated from relational table to XML document. The semantics of the relational database, captured in EER diagram, are mapped to XML schema using stepwise procedures. The physical data are then mapped to XML
document under the definitions of the XML schema.
All the researches above are Model-based Translation – which converts the relational schema to an intermediate model which then is mapped to XML using conversion rules. This research presents Flat translation algorithm which is depending on Visual Basic capabilities in transacting with data base and then converting all the relations between the tables and the fields to aggregation of arrays submitting to the algorithm steps to result XML file ready for opening with internet explorer.

2. Relational Data base to XML Translating algorithm

The algorithm begins with taking the data base as input and generates XML file as output. The processing steps of the algorithm are represented in the five steps below and for more simplicity and clarity, the research presents algorithm flowchart in figure 1.

1. Calculate the number of tables in the database and the number of primary and foreign keys in each table and build an array which contain them as follows:

- Connect to the data base by Visual Basic command

  Set dbs.opendatabase(path)

  Path is the location of the database in the computer

- Return the number of tables in the database

  Nooftables=dbs.tablescount

- Save the names of all tables of the data base in an array called **tb()**

For i=0 to Nooftables

Tb(i)=Dbs.tabledef(i).name

- Save all the primary keys in array called **primary()** with implementing the following condition inside the two iterations to all the fields of each table

For i=0 to nooftables

Nooffields=dbs.tabledef(i).fields.count

For j=0 to nooffields

If dbs.tabledef(i).fields(i).primary=true then

Primary(n)=dbs.tabledef(i).fields(i)

N=n+1

Endif

Next i

- Save all the foreign keys in array called **foreign()** by taking each primary key from primary array and search through the rest of the tables by **searchintable** function, if there is the same field name, the **searchintable** function returns true to allocate it inside the foreign array and if not the loop will continue to take the next element as represented below:

/ code for sending each elements to

/ seachintables function

For s=0 to n

Foreign=primary(s)

```
If searchintable(foreign) then

Foreign(no)=foreign

Endif

Next s

/ code for searchintables function

Public Function Searchintable(foreign)

For i=0 to nooftables

Set rst = dbs.OpenRecordset("SELECT * FROM " & tb(i))

rst.MoveLast

nf = rst.Fields.Count /no. of feields

For i = 0 To nf – 1

If foreign= rst.Fields(i).Name

Value=1 / the item is found

Exit function

End If

Next i

Value=0 / the item is not found

End Function
```

2. The algorithm build an array called **fields_names()** which contains all the names of the fields in all the tables in the database except the primary and foreign keys because they don't contain any values but contain information about how tables are linked so they will be attributes to our XML element.

3. The above operation done by linear search algorithm which translate from one table to another for finding all field names. the code steps below are inside an outer loop takes table names one after the other from **tb()** array and for each table the following steps are done:

```
For i = 0 To nf – 1

rs = searchid(rst.Fields(i).Name)

If rs = 1 Then

fields_names() (in1) = rst.Fields(i).Name

in1 = in1 + 1

End If

Next i
```

The above steps take the field names for each table and in each iteration it call a function called **searchid** to search inside the primary and foreign arrays and if it is found, it will not be inserted inside **fields_names()** array. The below steps are the **searchid** function code:

```
Public Function searchid(element)

Dim i, j As Integer

For i = 1 To nooftables – 1

For j = 0 To max_no_of_id – 1

If element = primary(i) or

element= foreign(i) Then

searchid = -1

Exit Function

Else

End If

Next j

Next i

searchid = 1

End Function
```

4. After retrieving all the names of the fields, the trip for translating them to tags in XML file will begin by the aid of two arrays; **fields_names()** (built in step2) and **fields_values()** (which will be built later), it will begin to find all the values of fields in all the records and translating them to values of tags, this will be done for all the records of all tables by multiple inner iterations and also by the aid of the previous search algorithm, the detailed steps are as follows:

- Open a new file for writing mode to write the XML converted code.
- Put some abbreviation for <,> ,/ and " as ot(open_tag), ct(close_tag), sl(slash) and dq(double quotation) respectively to use them in writing because they are principals in XML file.
- Write the database name as the first line in XML file to be the root

Document name closed with tags as follows.

```
e1 = ot & databasename & ct

Write #filenumber, e1
```

- Begin the first outer loop to take the first table name to be the first central element and write it with it's property (primary key and foreign key if exists) and the values of them as follows:

```
primarykey = rst1.Fields(0).Value

foreignkey = rst1.Fields(1).Value

e2 = ot & tb(i) & "  " & primary(i) & " = " &
     dq & primarykey & dq & " "  &
     foreign(i) & " = " & dq & forignkey
     & dq & ct

Write #filenumber, e2
```

this code is a part for large subroutine filled with many cases, in this case there is only one primary key and one foreign key (sometimes there are more than one foreign keys).

- Write fields names to be the child elements to the central element.

- Fill fields values by an inner loop which take the fields values from the current table for one record and after completing, the algorithm will write the information to XML file and returns to the outer loop to take the second table for the same record with new primary key value and new fields values, this operation is repeated until completing all the tables and this whole operation will be iterated for all the records of data base tables. This is the code for filling fields values array for one table:

```
For s = 0 To rst.Fields.Count – 1

If    IsNull(rst.Fields(s).Value)    Then
      fields_values(cs) = "" else

      fields_values = rst.Fields(s)

      endif

      next s
```

This is the code for writing fields names and values for one record to XML file:

```
                                For i = 0 To cs - 1

e3 = ot & fields_names(i) & ct &
fields_values(i) & ot & sl &
fields_names(i)  & ct

  Write #filenumber, e3

Next i
```

5. When the algorithm is terminated, the XML file will be ready to run by internet explorer to see the result.

All the code steps are implemented with Visual Basic. The research mentions the principal steps in the algorithm but it doesn't mention the variables declaration, controls declaration(like command button declaration and other controls) and the connection between controls because they will lead the reader to confusion

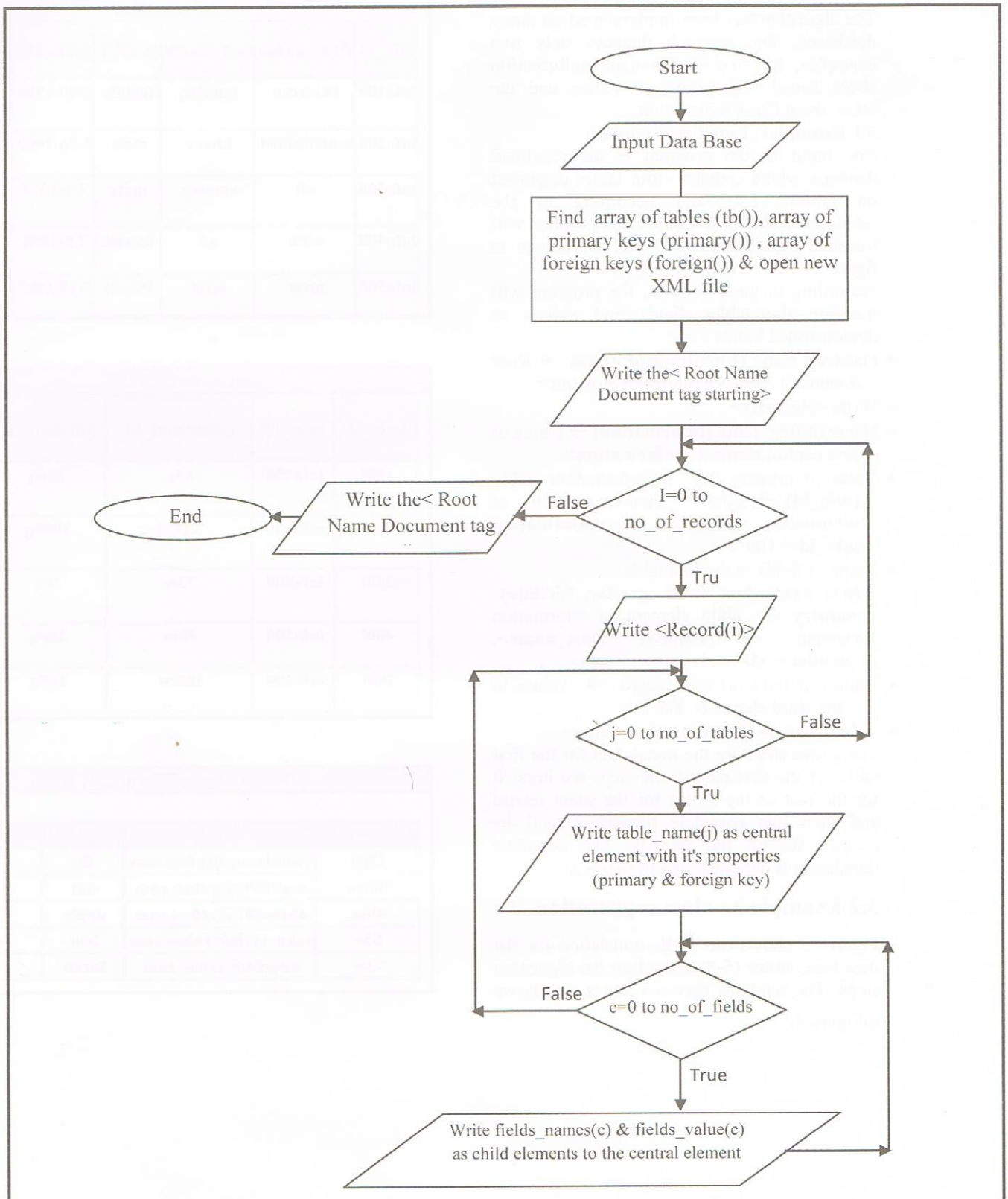

Figure 1:Algorithm Flowchart

3. Implementation of the algorithm

The algorithm has been implemented on many databases. The research displays only two examples, the first one contains information about Email registration of Yahoo and the other about Class registration.

3.1 Example1. Email registration:

The input to the program is the relational database which contains four tables displayed on tables (1-4) and according to the relationship shown in figure 2 the program will translate the database to XML file shown in figure 3 .

According to the algorithm, the program will translate the tables fields and values as demonstrated below :

- Database name **(Emailregisteration)** → Root document name **<Emailregisteration>**
- Write **<Record1>**
- Name of first table **(Information)** → Name of first central element **<information>**
- Value of primary key of information table **(info_id)** → value of element attributes of information element **<information info_id="100">**
- Name of fields of the first table **first_name**, **last_name**, **gender**, **birthday**, **country** → child element of information element **<first_name>**, **<last_name>**, **<gender>**, **<birthday>**, **<country>**
- Values of fields of first record → values to the child elements like this
- <first _name> **yasmeen**</first_name>

The above steps are the translation for the first table for the first record, the steps are iterated for the rest of the tables for the same record and then this complete translation will be iterated for all the records. The complete translation is demonstrated in figure 3.

3.2 Example 2 : class registration

Figures 5 shows the XML translation for the data base tables (5-8) according the algorithm steps. The relations between tables are shown in figure 4.

Table 1: Information table

information					
info_id	first_name	last_name	gender	birthday	country
info100	yasmeen	muafaq	female	8/20/1980	iraq
info200	mohammed	khary	male	5/26/1969	iraq
info300	ali	sameer	male	1/1/1978	egypt
info400	suha	ali	female	7/5/1990	syria
info500	noor	seret	female	7/19/1987	london

Table 2: link_table

link_id	info_id	password_id	question_id
100l	info400	65c	256q
200l	info100	130gf	1000q
250l	info500	73w	70q
400l	info300	40m	200q
700l	info200	300sw	105q

Table 3:password_info table

password_id	email	password	re_type_password
130g	yasminmq@yahoo.com	fly	fly
300sw	moh2000@yahoo.com	star	star
40m	ahmed87@yahoo.com	apple	apple
65c	suha_syria@yahoo.com	lion	lion
73w	noor86@yahoo.com	farah	farah

Table 4: questio_info table

question_id	alternet_email	secretQ1	youranswer1	secretQ2	youranswer2
1000q	yasminmq80@yahoo.com	what's your youngest child's nickname?	amoony	what's your oldest child's nickname?	barhoomy
105q	mohalshamaa@yahoo.com	what's your favorite car?	corella	where did you spend your honu moon?	jordan
200q	ah_87@yahoo.com	what's your oldest child's nickname?	ameer	what's the name of your favorite book?	visual_basic6
256q	su_syria@yahoo.com	what's your favorite auther?	ahmed_ismaeel	what's your favorite car?	corella
70q	light86@yahoo.com	what's your youngest child's nickname?	nono	what's your favorite car?	corella

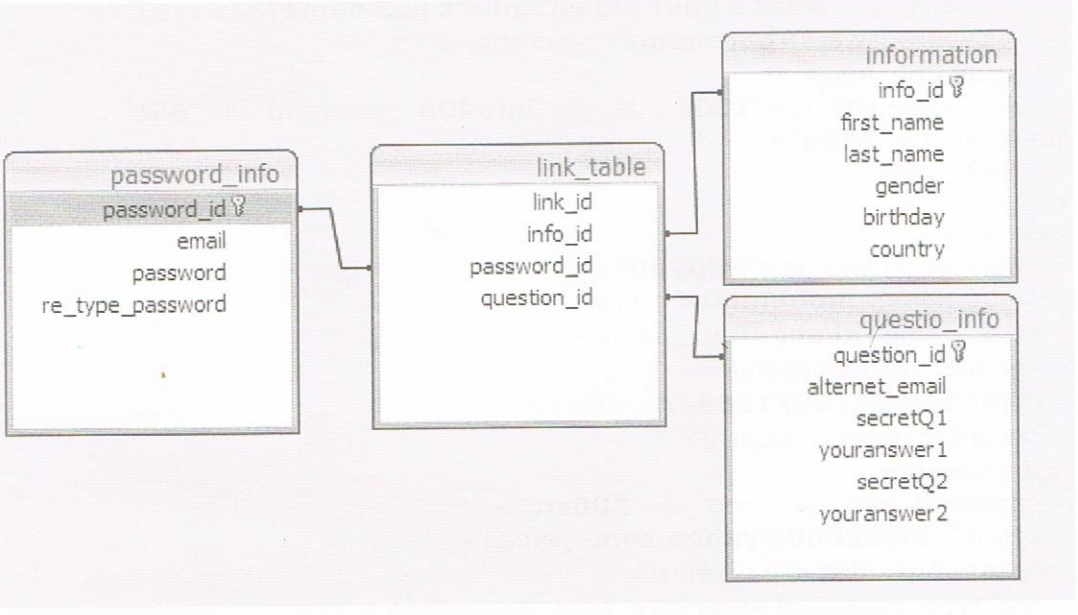

Figure 2
Relationship between tables in Email Registeration Database

```
    emailregisteration>
  <R ecord1>
   <information info_id="info100">
     <first_name>yasmeen</first_name>
     <last_name>muafaq</last_name>
     <gender>female</gender>
     <birthday>20/08/1980</birthday>
     <country>iraq</country>
   </information>
   <password_info password_id="130g">
     <email>yasminmq@yahoo.com</email>
     <password>fly</password>
     <re_type_password>fly</re_type_password>
   </password_info>
     <question_info question_id="1000q">
      <alternet_email>yasminmq80@yahoo.com</alternet_email>
      <secretQ1>what's your youngest child's nickname?</secretQ1>
      <youranswer1>amoony</youranswer1>
      <secretQ2>what's your oldest child's nickname?</secretQ2>
      <youranswer2>barhoomy</youranswer2>
   </question_info>
   <link_table link_id="100i" info_id="info400" password_id="65c"
question_id="256q">
   </link_table>
</Record1>
<Record2>
 <information info_id="info200">
   <first_name>mohammed</first_name>
   <last_name>khary</last_name>
   <gender>male</gender>
   <birthday>26/05/1969</birthday>
   <country>iraq</country>
 </information>
 <password_info password_id="300sw">
   <email>moh2000@yahoo.com</email>
   <password>star</password>
   <re_type_password>star</re_type_password>
</password_info>
<question_info question_id="105q">
   <alternet_email>mohalshamaa@yahoo.com</alternet_email>
   <secretQ1>what's your favorite car?</secretQ1>
   <youranswer1>corella</youranswer1>
   <secretQ2>where did you spend your honu moon?</secretQ2>
   <youranswer2>jordan</youranswer2>
</question_info>
<link_table link_id="200i" info_id="info100" password_id="130gf"
question_id="1000q">
 </link_table>
</Record2>
<Record3>
<information info_id="info300">
   <first_name>ali</first_name>
   <last_name>sameer</last_name>
```

```
    <gender>male</gender>
    <birthday>01/01/1978</birthday>
    <country>egypt</country>
  </information>
  <password_info password_id="40m">
    <email>ahmed87@yahoo.com</email>
    <password>apple</password>
    <re_type_password>apple</re_type_password>
  </password_info>
    <question_info question_id="300q">
    <alternet_email>ah_87@yahoo.com</alternet_email>
    <secretQ1>what's your oldest child's nickname?</secretQ1>
    <youranswer1>ameer</youranswer1>
    <secretQ2>what's the name of your favorite book?</secretQ2>
    <youranswer2>visual_basic6</youranswer2>
  </question_info>
  <link_table link_id="250" info_id="info500" password_id="73w"
  question_id="70q">
  </link_table>
  </Record3>
  <Record4>
  <information info_id="info400">
    <first_name>suha</first_name>
    <last_name>ali</last_name>
    <gender>female</gender>
    <birthday>05/07/1990</birthday>
    <country>syria</country>
  </information>
  <password_info password_id="65c">
    <email>suha_syria@yahoo.com</email>
    <password>lion</password>
    <re_type_password>lion</re_type_password>
  </password_info>
    <question_info question_id="256q">
    <alternet_email>su_syria@yahoo.com</alternet_email>
    <secretQ1>what's your favorite auther?</secretQ1>
    <youranswer1>ahmed_ismaeel</youranswer1>
    <secretQ2>what's your favorite car?</secretQ2>
    <youranswer2>corella</youranswer2>
  </question_info>
  <link_table link_id="400" info_id="info300" password_id="40m"
  question_id="200q">
  </link_table>
  </Record4>
  <Record5>
  <information info_id="info500">
    <first_name>noor</first_name>
    <last_name>seret</last_name>
    <gender>female</gender>
    <birthday>19/07/1987</birthday>
    <country>london</country>
  </information>
  <password_info password_id="73w">
     <email>noor86@yahoo.com</email>
```

```
    <password>farah</password>
    <re_type_password>farah</re_type_password>
  </password_info>
    <question_info question_id="70q">
    <alternet_email>light86@yahoo.com</alternet_email>
    <secretQ1>what's your youngest child's nickname?</secretQ1>
    <youranswer1>nono</youranswer1>
    <secretQ2>what's your favorite car?</secretQ2>
    <youranswer2>corella</youranswer2>
   </question_info>
  <link_table link_id="700" info_id="info200" password_id="300sw"
  question_id="105q">
  </link_table>
  </Record5>
    </emailregisteration>
```

Figure 3
Translated XML File for Email Registeration

Table 5

class

class_id	course_id	location	date	time_start	time_end
c100	w100	nahrainuni	9/2/2010	12:00:00 PM	3:00:00 AM
c200	w200	baghdaduni	9/10/2010	12:00:00 PM	3:00:00 AM
c300	w300	nahrainuni	9/20/2010	3:00:00 AM	6:00:00 AM
c400	w400	tecknologyuni	10/1/2010	12:00:00 PM	3:00:00 AM

Table 6

class_student

class_student_id	class_id	student_id
a1	c100	1
a2	c200	2
a3	c300	3
a4	c400	4

Table7

course

course_id	course_name
w100	programming
w200	database
w300	xml
w400	operatingsystem

Table 8

student

student_id	first_name	last_name	organization	phone	email
1	yasmeen	muafaq	nahrain	55674500	yasmeen@yahoo.com
2	uday	muafaq	baghdad	55677720	uday@yahoo.com
3	zaid	rafea	baghdad	88356890	zaid@yahoo.com
4	mazin	zuher	tecknology	5551002	mazin@yahoo.com

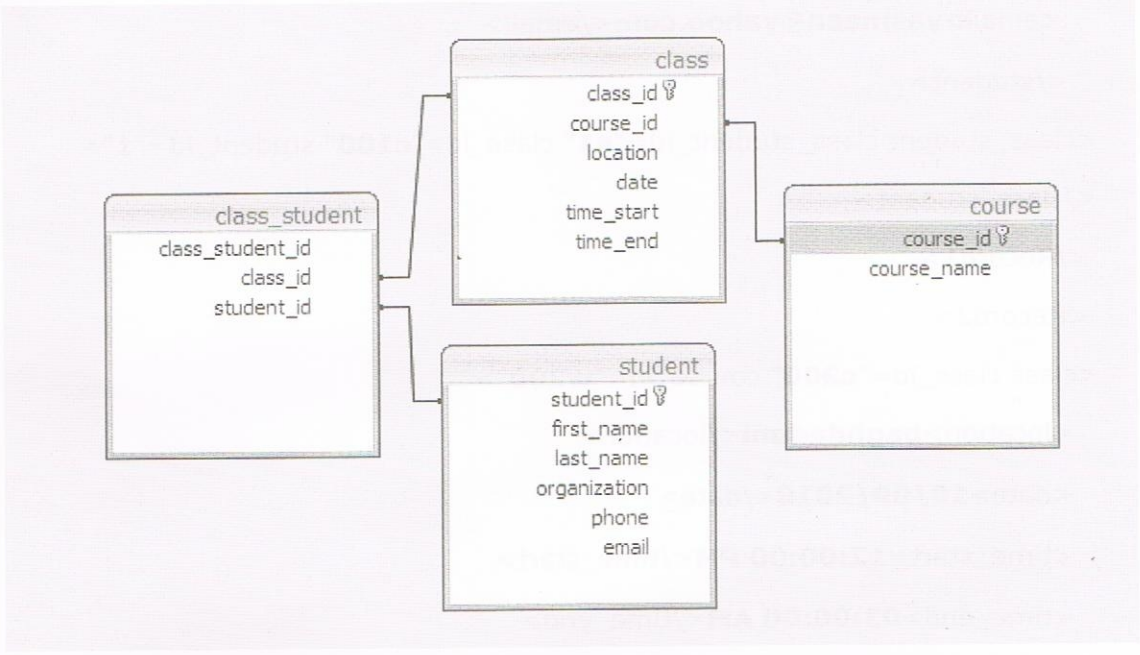

Figure 4
Relationship between tables in Class Registration Database

```
<Classregisteration>
  <Record1>
   <class class_id="c100" course_id="w100">
    <location>nahrainuni</location>
    <date>02/09/2010</date>
    <time_start>12:00:00 PM</time_start>
    <time_end>03:00:00 AM</time_end>
  </class>
  <course course_id="w100">
    <course_name>programming</course_name>
  </course>
  <student student_id="1">
    <first_name>yasmeen</first_name>
    <last_name>muafaq</last_name>
    <organization>nahrain</organization>
```

```
    <phone>55674500</phone>
    <email>yasmeen@yahoo.com</email>
   </student>
<class_student class_student_id="a1" class_id="c100" student_id="1">
</class_student>
 </Record1>
 <Record2>
<class class_id="c200" course_id="w200">
  <location>baghdaduni</location>
  <date>10/09/2010</date>
  <time_start>12:00:00 PM</time_start>
  <time_end>03:00:00 AM</time_end>
</class>
<course course_id="w200">
  <course_name>database</course_name>
</course>
<student student_id="2">
  <first_name>uday</first_name>
  <last_name>muafaq</last_name>
  <organization>baghdad</organization>
  <phone>55677720</phone>
  <email>uday@yahoo.com</email>
 </student>
<class_student class_student_id="a2" class_id="c200" student_id="2">
</class_student>
 </Record2>
 <Record3>
  <class class_id="c300" course_id="w300">
  <location>nahrainuni</location>
  <date>20/09/2010</date>
```

```
  <time_start>03:00:00 AM</time_start>
  <time_end>06:00:00 AM</time_end>
</class>
<course course_id="w300">
  <course_name>xml</course_name>
</course>
<student student_id="3">
  <first_name>zaid</first_name>
  <last_name>rafea</last_name>
  <organization>baghdad</organization>
  <phone>88356890</phone>
  <email>zaid@yahoo.com</email>
  </student>
<class_student class_student_id="a3" class_id="c300" student_id="3">
</class_student>
 </Record3>
 <Record4>
<class class_id="c400" course_id="w400">
  <location>tecknologyuni</location>
  <date>01/10/2010</date>
  <time_start>12:00:00 PM</time_start>
  <time_end>03:00:00 AM</time_end>
</class>
<course course_id="w300">
  <course_name>operatingsystem</course_name>
 </course>
<student student_id="4">
  <first_name>mazin</first_name>
  <last_name>zuher</last_name>
  <organization>tecknology</organization>
```

```
<phone>5551002</phone>
<email>mazin@yahoo.com</email>
</student>
<class_student class_student_id="a4" class_id="c400" student_id="4">
</class_student>
</Record4>
</Classregisteration>
```

Figure 5
Translated XML File for Class Registeration

4. Conclusion

Relational databases are the dominant data storage model and XML the de facto standard for sending information over the Internet. Being able to transfer data between the two formats has become an important task. This research has presented an algorithm for translating data from RDBMS to XML database depending on visual basic capabilities in connecting with database and it's ability in characterizing all it's connections and properties to extract all these information in a set of temporary arrays and begin to convert them in XML format by multiple inner iterations. The XML format is the algorithm output and it is ready to run by internet explorer.

References

1. Jayavel S. et al.: "Relational databases for querying XML documents: Limitations and opportunities", In VLDB Conference, 1999.

2. Jayavel S. et al.: "Efficiently publishing relational data as XML documents", VLDB Journal: Very Large Data Bases, 2001.

3. Gavin P.: "Beginning XML Databases", Wiley Publishing, Inc., Indianapolis, Indiana, Canada, 2007.

4. Jayavel S. et al.: "A general techniques for querying XML documents using a relational database system.", SIGMOD Record, 2001.

5. Chunyan W. et al.: " Converting Legacy Relational Database into XML Database through Reverse Engineering", http://pages.cpsc.ucalgary.ca/~alhajj/papers/p25.pdf, 2004.

6. Ramon L.: " A Cost-Based Approach for Converting Relational Schemas to XML". www.cs.uiowa.edu,2010.

7. Joseph F. et al.: " Converting Relational Database into XML Document", IEEE Computer Society Washington, DC, USA, 2001

8. Florescu D. et al.: "Storing and Querying XML Data Using an RDBMS", IEEE Data Engineering Bulletin 22, 1999.

تصميم و تنفيذ مترجم من قاعدة البيانات العلائقية إلى لغة ال (XML)

ياسمين موفق قاسم
مدرس مساعد
كلية الهندسة/جامعة النهرين

خلاصة

حديثا ال XML هي الشكل الشائع لتمثيل و خزن و تبديل و الوصول الى البيانات و حاليا تتعرض الى تحدي مقارنة بقاعدة البيانات لشيوعها و كثرة استعمالها و استمرارية خزنها للبيانات يوميا لذلك فمن المتوقع أن تكون هناك احتياجات للتحويل من قاعدة البيانات العلائقية الى ال (XML),هذا البحث يعرض خطوات لهذا التحويل قد نفذت بواسطة لغة الفيجوال بيسك للحصول على مترجم يترجم من قاعدة البيانات ذي العلاقة الأحادية إلى (XML).

البرنامج أو المترجم يطلب قاعدة البيانات كعنصر إدخال و ينتج ال (XML) كعنصر إخراج, المترجم يقوم بتحويل اسم قاعدة البيانات إلى النود الرئيسي في ال (XML) فايل و الذي يضم جميع العناصر الأخرى ثم يقوم بتحويل اسم الجدول الرئيسي إلى اسم العنصر الرئيسي و من ثم جميع أسماء الحقول إلى أسماء العناصر ضمن العنصر الرئيسي و ينقل القيم الموجودة في كل سجل إلى قيم العناصر التي تقابلها و لن ينسى بتحويل المفتاح الرئيسي للجدول إلى خصائص لهذا الجدول . قد نفذ هذا المترجم على كثير من الجداول قد ذكرنا منها اثنان فقط وهي (تسجيل البريد الالكتروني) و (تسجيل الصف) مع عرض نتائج المترجم.

قاعدة البيانات بواسطة إل(XML) تفتح آفاق جديدة في المعالجة و صنع الكويري, تكرار المحتوى قد انحصر و قد أدى إلى تماثل جيد للمعلومات بالإضافة إلى أن البحث على المعلومة قد تحسن مما أدى إلى استرجاع المعلومات بشكل سريع و مضمون .