The Libyan Journal of Science (An International Journal): Volume 21A, 2018

Migration of RDBs into ORDBs and XML Data

Ali Elbekai[1], Abduelbaset Goweder[2], and Abdulhafid Shuwehdi[3]

[1]National Board for Technical & Vocational Education, Tripoli, Libya
ali.elbekai@gmail.com
[2]Computer Science Department, The Libyan Academy, Tripoli, Libya
agoweder@yahoo.com
[3]Computer Science Department, Tripoli University, Tripoli, Libya
shwuehdi@yahoo.com

Abstract

XML and relational database are two of the most important mechanisms for storing and transferring data. A reliable and flexible way of moving data between them is a very desirable goal. The way data is stored in each method is very different which makes the translation process difficult. To try and abstract some of the differences away, a low–level common data model can be used to successfully move data from one model to the other. A way of describing the schema is needed. To the best of our knowledge, there is no widely accepted way of doing this for XML. Recently, XML Schema has taken on this role. On one hand, this study takes XML conforming to XML schema definitions and transforms them into relational database via the low–level modeling language HDM. On the other hand, a relational database is transformed into an XML Schema document and an XML instance document containing the data from the database. The transformations are done within the Auto med framework providing a sound theoretical basis for this work. A visual tool that represents the XML Schema in tree structure and allows some manipulation of the schema is also described.

Keywords: Data migration; XML; RDBM; ORDBM.

المستخلص

إن XML وقواعد البيانات العلائقية أليتان من الآليات الأكثر أهمية في تخزين ونقل أو تحويل البيانات. إن الطريقة المرنة والتي يعول عليها في تحويل البيانات بين هاتين الآليتين هي هدف مرغوب جداً. الطريقة التي تخزن بها البيانات في كل منهج مختلفة مما يجعل عملية الترجمة صعبة. لمحاولة فصل بعض الإختلافات يمكن إستخدام نموذج بيانات مشترك منخفض المستوي لتحويل البيانات من نموذج إلي آخر بنجاح. الحاجة ماسة إلي طريقة لوصف المخطط. حسب علمنا لاتوجد طريقة مقبولة بشكل كبير جداً بالنسبة لـ XML. حديثاً, مخطط XML قد أخذ هذا الدور. من ناحية تأخذ هذه الدراسة XML المطابقة لتعريفات مخطط XML وتحولها إلي قاعده بيانات علائقية عن طريق لغة النمذجة HDM المنخفضة المستوي. ومن ناحية أخري تحول قاعدة البيانات العلائقية إلي وثيقة مخطط XML ووثيقة XML تحتوي على البيانات من قاعدة البيانات. تجري التحويلات في ما يعرف Auto med framework موفراً قاعدة نظرية سليمة للعمل. الأداة المرئية التي تمثل مخطط XML في هيكلية الشجرة والتي تسمح بإجراء بعض المعالجة للمخطط يتم وصفها كذلك.

Accepted for publication: 22/4/2018

INTRODUCTION

Many organizations have stored their data in RDBs and wish to take advantage of databases that have emerged more recently, such as object relational and XML document. Instead of discarding the previous RDB or build non-relational applications on it, it is accepted to convert the old data and applications together into a new environment.

The increasing popularity of new object-based and World Wide Web (WWW) technologies and non-traditional applications e.g., multimedia, geographical information systems, computer aided design, etc. can be considered the most significant recent changes in information technology.

These novel technologies have been dominant in the area of information systems due to their productivity, flexibility and extensibility. Object-Relational Databases (ORDBs) and the eXtensible Markup Language (XML), which all support various concepts, have been proposed in order to fulfill the demands of complex applications that require rich data types. Consequently, new types of DBMSs have started to emerge in the market providing more functionality and flexibility. However, as the majority of databases are still stored and maintained in RDBMSs, it is therefore expected that the need to convert such RDBs into the technologies that have emerged recently will grow substantially [1,2,3,4,5,6,7].

The need for additional data modelling feature has also been recognized by RDBMS vendors, leading to the appearance of object-based relational systems [8,9,10,11,12,13]. An ORDB has potential because it has a relational technology base and object features. The main goal of its design was to incorporate both the strong transaction and performance management features of the relational model and the features of flexibility, scalability and support for rich data types of OO models.

Also transforming data from one format to another is frequently required in modern information systems and Web applications that need to exchange or integrate data. As XML becomes the standard for data exchange among applications (over the Web), transformation of XML data also becomes increasingly important. XML is useful for modelling data of all types. It can be created and altered with any text editor [14,15]. XML is extensible and platform-independent, which makes it very useful technology in achieving compatibility between different programming platforms and operating systems.

This study provides an algorithm as a solution; it is a model for migrating from RDBs into object relational and XML data structure.

OVERVIEW OF RDB, ORDB AND XML TECHNOLOGIES

This section provides an overview on the main concepts of relational, object-relational and XML data, in order to get a better understanding of the process of RDB migration. So, an introduction of the basic concepts of the data models relevant to the database migration process and relational data model are introduced.

A DBMS is a shell surrounding one or more databases throughout various interactions such as data maintenance, data retrieval, and data control. DBMS is a set of programs that enables storing, modifying, and extracting information from a database, it also provides users to add, delete, access, modify, and analyze data stored in a database. There are several DBMS according to the data model used in database design. Each data model has its own way of storing the data [10]. The sequential order of the devolvement of DBMS is Hierarchical, Network, Relational, object oriented and object relational.

RELATIONAL DATABASE (RDB)

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by Codd in 1969 [8]. Where the data are organized as tables of data values (relation) that are related to each other, it is a collection of one or more relations, where each relation is a table with rows and columns, and all the operations on the data done on the these tables.

In relational database systems the data and the information resulting from a query are represented using Tables (relations). A table is a two dimensional array uniquely identified by its name and consists of rows and columns. Each row containing exactly one tuple or record, which contains stored data related to an entity such as a student and each column contains the data related to a single attribute of the entity such as student name. A table can have one or more columns, a column is made up of a column name and a data type, and it describes an attribute of the tuples [9]. The type of data to be stored in a table is defined by the data types of the attributes at table creation time [11]. The structure of a table, also called relation schema, is thus defined by its attributes. There is a rule for a table to be relation as follows:

- The intersection of row with the column should contain single value (atomic value).
- All entries in a column are of same type.
- Each column has a unique name (column order not significant).
- No two rows are identical (row order not significant)
- Relations must have a key which can be a set of attributes.
  The key is an attribute or group of attributes, which is used to identify a row in a relation. The key can be classified into:
- ▪ Primary Key: a set of columns used to uniquely identify rows of a table. The primary key should not be null.
- ▪ Foreign Key: is set of fields or attributes in one relation that is used to refer to a tuple in another relation.
- ▪ Composite Key: is a primary key that is made up of more than one attribute. In some tables a single column cannot be used to uniquely identify entities (rows), in that case two or more columns are used to uniquely identify rows of the table. When a primary key contains two or more columns it is called as composite primary key.

An integrity constraint (IC) is a condition that is specified on a database schema, and restricts the data that can be stored in an instance of the database.

Data integrity constraints refer to the accuracy and correctness of data in the database, A DBMS enforces integrity constraints to prevent the entry of incorrect information and it permits only legal instances to be stored in the database. If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance.Data integrity maintains data consistency for operations such as insert, update, and delete [10]. Types of data integrity constraints are classified as:

- Domain Integrity (DI) constraints are used to specify the valid values that a column defined over the domain can take [10].
- Entity Integrity (EI) specifies that all values in primary key must not be null and unique [10].
- Referential Integrity (RI) specifies a column or a list of columns as a foreign key of the referencing table. The referencing table is called the child-table, and the referenced table is called the parent - table. In other words, one cannot define a referential integrity constraint that refers to a table R before that table R has been created. This specifies that a foreign key must be either null or must have a value that is derived from corresponding parent key. As example, if we have a table called BATCHES, then ROLLNO column of the table will be referencing ROLLNO column of STUDENTS table. All the values of ROLLNO column of BATCHES table must be derived from ROLLNO column of STUDENTS table. So that no student who is not part of STUDENTS table can join a batch.

OBJECT RELATIONAL DATABASE MANAGEMENT SYSTEM (ORDBMS)

Object relational database ORDBMs is the third type of database common today. The object-relational technology is relatively recent, but it is not a new technology on its own right. It is also known as enhanced relational database systems (ERDBMs) or Extended relational. As it extends a relational model to include Object Oriented (OO) capabilities. ORDBMs are systems that attempt to merge relational database systems with object oriented database concept: objects, classes and inheritance are directly supported in database schemas and in the query language, it is a database that adds features associated with object oriented systems to RDBMs.

The main objective of their design was to incorporate both the performance management features of RDBs, and the OO model features of flexibility, scalability and support for rich data types, together with the ability to handle alphanumerical data types, ORDBs were created to handle multimedia data types such as audio, video and image files that relational database were not equipped to handle [12].

Developers can work with tabular relational structures and DDL with the possibility of object management. Furthermore, the development of ORDBs was the result of increased usage of Object Oriented Programming Languages to avoid the mismatch between these languages and DBMSs.

Many vendors have created their SQL languages to be used for their own products to capture the OO concepts. This has led to a lack of standards that can be used by all ORDB users, and the differences between these products may be significant. Therefore, the ANSI and ISO SQL standardization developed a standard called SQL-1999 (or SQL3) to support object-oriented data management, which became the foundation for most ORDBMs. SQL3 and its successor SQL4 are extensions of SQL-92 to include the new features of OO concepts. SQL3 supports ADTs, including OIDs, methods, inheritance, polymorphism [12,13].

EXTENSIBLE MARKUP LANGUAGE (XML)

XML is a markup language. It is not a programming language but instead is a text based format which provides mechanism for describing document structures. XML is an open standard, not owned by any one organization. This is W3C recommendation. XML is known as the next generation of HTML and a simplified subset of the Standard Generalized Markup Language (SGML) similar to HTML. XML was designed to transport and store data [15]. SGML is the parent of all markup languages it was too hard to implement and too large for presenting structured documents on the web (it is complex). HTML is a markup language that is designed to present information on the web but does not have the ability to store data and metadata; HTML was designed to display data [14]. XML which is a subset of SGM, takes the best features of SGML allowing it to mark up data in a standard, generalized way, but strips out the complexities that make SGML hard to implement. XML and its related technologies are emerging as the standard in data storage, transmission and manipulation. XML was designed to carry data, not to display data. The XML documents are a collection of text documents with additional tags and relations between these tags. These documents can be opened with any program that knows how to read a text file and it is designed to be self-descriptive. XML tags are not predefined and it allows the author to define his/her own tags and his/her own document structure.

The main difference between HTML and XML is that in HTML the semantics and syntax of tags is fixed. In XML the author of the document is free to create tags whose syntax and semantics are specific to the target application. Also the semantics of a tag are not tied down but is instead dependent on the context of the application that processes the document. XML is the most common tool for data transmissions between all sorts of applications.

XML documents are composed of markup and content. There are several kinds of markup that can be noted in an XML document: prolog, elements, comments, entity references and marked section. Each of these markup concepts are described as follows.

The top of an XML document is graced with special information called document prolog. At its simplest, the prolog merely says that this is an XML document and declares the version of XML being used: <?xml version="1.0"?>.

The XML declaration is an announcement to the XML processor that this document is marked up in XML. Its forms are shown as follows:

```
<?xml name = "val1" name2 = "val2" ?>.
An example of well-formed XML declaration.
<?xml version="1.0" encoding="us-ascii" Standalone= "yes"?>
<?xml version="1.0" encoding="ISO-8859-
1"Standalone="no"?>
```

The second part of the prolog is the document type declaration (DTD). A DTD is a set of rules that defines what tags appear in XML document, what attributes the tags may have and what relationship the tags have with each other. When an XMLdocument is processed, it is compared within the DTD to be sure it is structured correctly and all tags are used in the proper manner.

DTD defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes. A DTD can be declared inline inside an XML document, or as an external reference. If the DTD is declared inside the XML file, it should be wrapped in a DOCTYPE definition with the following syntax.

<!DOCTYPE root-element [element-declarations

Elements are the most common form of markup. An XML element is everything from the element's start tag <element> to the element's end tag <element>. An element can contain: other elements, text, attributes or a mix of all of all these. Fig. 1 is an example of XML document, it can be noticed that <bookstore> and <book> have element contents, because they contain other elements. <book> also has an attribute (category="CHILDREN"). <title>, <author>, <year>, and <price> have text contents [16,17].

The XML schema is an XML-based alternative to DTD. It is a set of recommendations from W3C. The XML schema describes the structure, content and semantic of an XML instance documents or XML document. With XML schema definition, it is possible to model how the data in the XML instance document is to be represented and how data is related to each other, i.e. parent / child. The XML schema definition language is also referred to as XML Schema Definition (XSD). And it makes it possible to define the data types and is extensible because it is composed of XML-syntax.

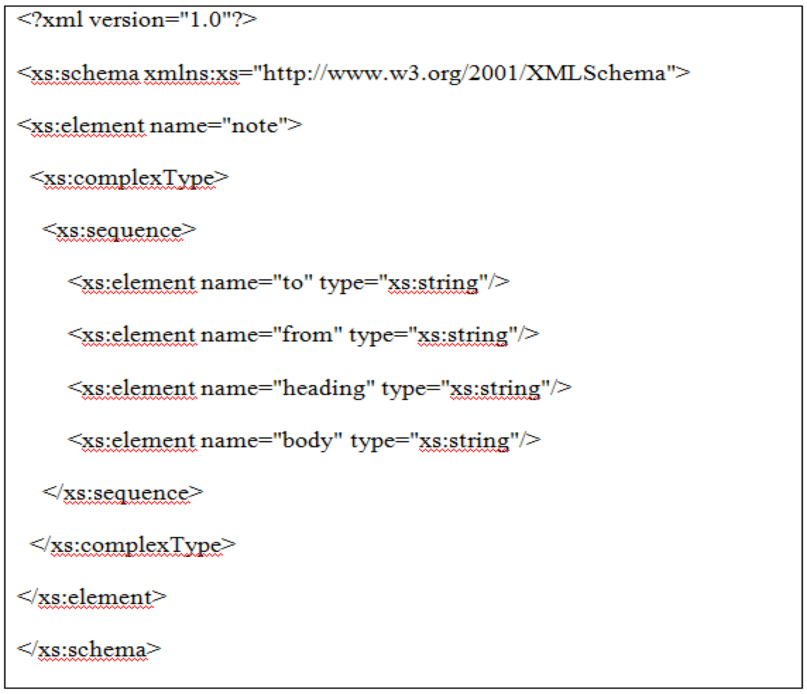
The predecessor DTD definition language, is composed of non- XML syntax. hence it is non- extensible which implies that it will constrain the evolution of XML. The XML schema definition language has support for namespaces while DTD has not. XML schema definition language is a key component of web services specifications such as SOAP8, and is widely used to describe XML vocabularies precisely. Fig. 2 shows a sample of an XML Schema.

Fig. 2 presents an XML Schema file called "note.XSD" that defines the elements. The note element is a complex type because it contains other elements. The other elements (to; from; heading; body) are simple types because they do not contain other elements [18,19,20].

```
<bookstore>
        <book category="CHILDREN">
             <title>Harry Potter</title>
             <author>J K.  Rowling</author>
             <year>2005</year>
             <price>29.99</price>
        </book>
  </bookstore>
```

Fig. 1. Sample of XML document.

In addition, there are many advantages to using XML for information exchange, and they offer many benefits to the user. XML is an extremely portable language to the extent that it can be used on large networks with multiple platforms like the internet, and can be used on handhelds or palmtops or PDAs. The XML is an extendable language, meaning that you can create your own tags, and it is as easy as HTML. It is a platform independent language and it can be combined with any application which is capable of processing XML regardless of the platform [14,15].



```xml
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="note">

 <xs:complexType>

  <xs:sequence>

    <xs:element name="to" type="xs:string"/>

    <xs:element name="from" type="xs:string"/>

    <xs:element name="heading" type="xs:string"/>

    <xs:element name="body" type="xs:string"/>

  </xs:sequence>

 </xs:complexType>

</xs:element>

</xs:schema>
```

Fig. 2. Sample of an XML Schema.

ALGORITHM DESIGN FOR MIGRATION PROCESS

This section presents and describes the design process of the algorithm that can be used for Migration process from Relational Database Management system to object Relational Database Management system and XML. The problem was how to migrate the RDBMS as a source into the newer databases as a targets and what is the best way to do that?

In this algorithm, Dataset concept is used. This Dataset is an object in .Net environment that contains data tables where used data application is temporarily stored. The design process of this algorithm consists of two phases: dataset creation and conversion process.

ANALYSIS OF THE ALGORITHM

The input of the algorithm is:

Relational Database System (SQL-Access).

The output of the algorithm will be:-

- XML File.
- Object Relational DMS (Oracle).

DESCRIPTION OF THE ALGORITHM

This section will describe the algorithm for mapping from RDBMS into ORDBMS and XML files. The main idea of this algorithm is based on two steps: First step is dataset creation which indicates how to get all the relational database contents and put it in a dataset. Second step is conversion process from dataset to our targets.

Throughout the design stage of this algorithm, several queries for dataset creation are used including the following:

1. A first query returns all tables names.
2. A second query gets the structure of each table and creates or adds these tables on a dataset using some functions.
3. A third query obtains the relations among all these tables then inserts these relations into the dataset.

When the dataset is prepared with data and relations, it needs be converted to ORDBMS and XML. In the conversion phase, there are some methods in .Net to convert dataset content into XML such as WriteXML/ WriteXML schema. So, these functions will be used to get XML document. To convert the dataset content into ORDBMS, the contents of the dataset are retrieved and a command statement is created and executed to obtain the objects on oracle format, then the relations are added to the dataset.

HOW THE ALGORITHM WORKS

Fig. 3 shows the diagram of the migration process which starts by selecting a database type and name. Next, the test connection to a database is tested by sending a message, if the database is disconnected, the algorithm will display a message saying: Connection Failed, and tries repeatedly until the connection is set and a message saying: “Connection Successful” is displayed. Then, a dataset is created and all database contents are added, after that the conversion to XML is done and consequently XML file is constructed. Also, the conversion to ORDBMS will be achieved and the result will be OR schema and objects on oracle format.

IMPLEMENTATION OF THE ALGORITHM

The implementation of the algorithm will be divided into three parts. First part is a connection between the application and the source and target is database. Second, retrieving the source database content to create the dataset. Third, the conversion process to XML and ORDBMS.

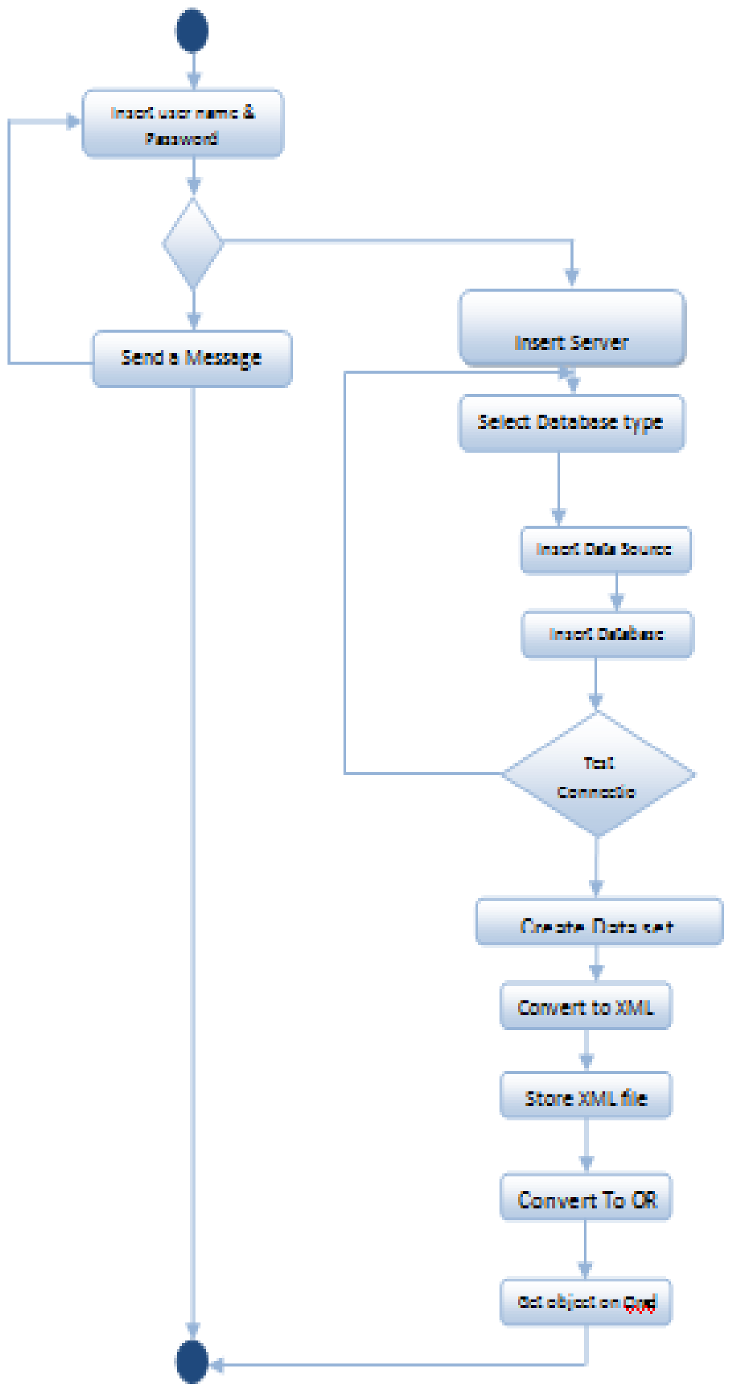


Fig. 3. A diagram of the algorithm.

SYSTEM ARCHITECTURE

The proposed system architecture is self-describing as shown in Fig. 4.

THE SYSTEM INTERFACE

The proposed system is implemented using the programming language VB.NET. Access and SQL SERVER 2005 are used for the inputs. The ORACLE 11g is used for the outputs of ORDB and the XML browser for the XML document. The interface system consists of three forms as shown in Fig. 5(a, b, and c). Fig. 5a shows a login window which displays the project's name and requests the login name and password of the system user. Fig. 5b shows the second form the "Server Configuration" which asks the user to insert the Oracle server name and Oracle user name, and password. Fig. 5c shows the third form, "The main form", of the conversion process.

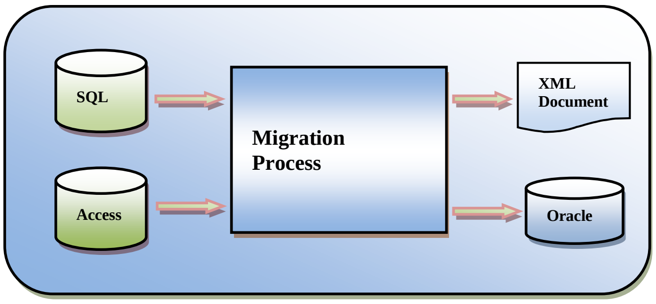


Fig. 4a. The proposed system architecture.

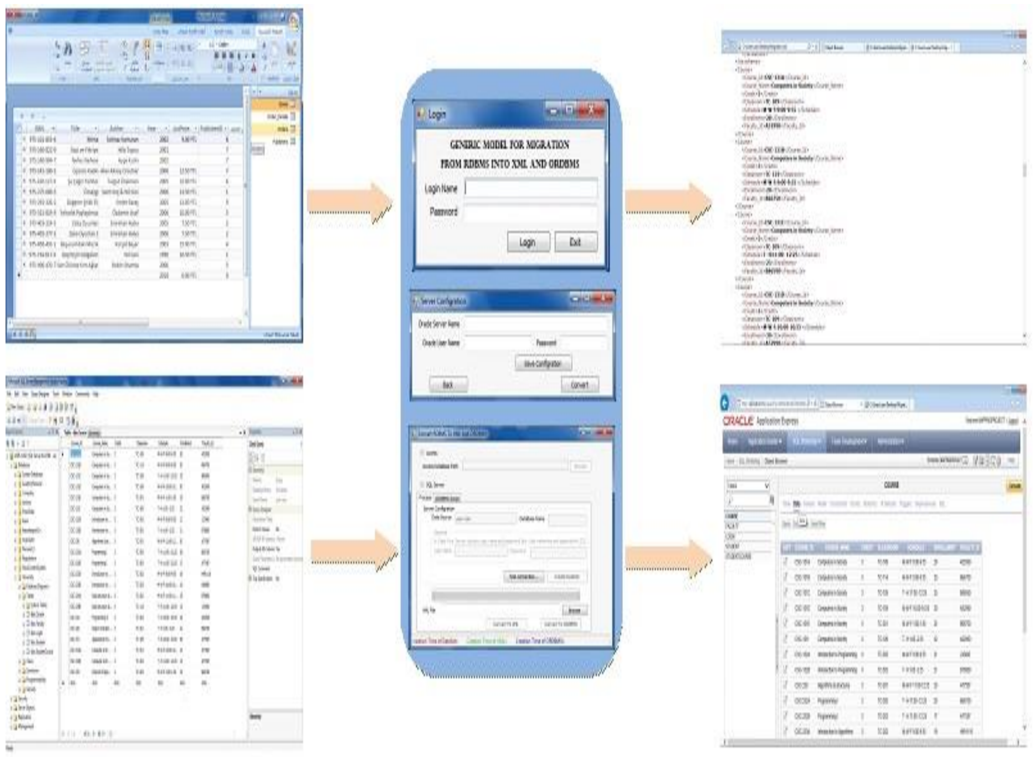


Fig. 4b. The proposed system architecture.

In this window, first it is required to enter the database source. Secondly "Test Connection" button can be clicked to make sure the connection among the application, the source, and target database is established. Then the "Create Dataset" button would enable you to create the dataset then the application would be ready for executing conversion buttons according to the required goal.

EVALUATION

In order to make the proposed system ready for use, it should be evaluated to make sure that the system fulfils the proposed objectives and demonstrates the operations of the software.

This system is built for mapping the RDB into XML data and ORDB. It has a range of functions and procedures in the process of reading and retrieving the RDB tables and converting them into objects and storing these objects and their constraints and data into ORDBMS. Also, it converts all database tables into XML element with all standards and data and stores it in XML document.

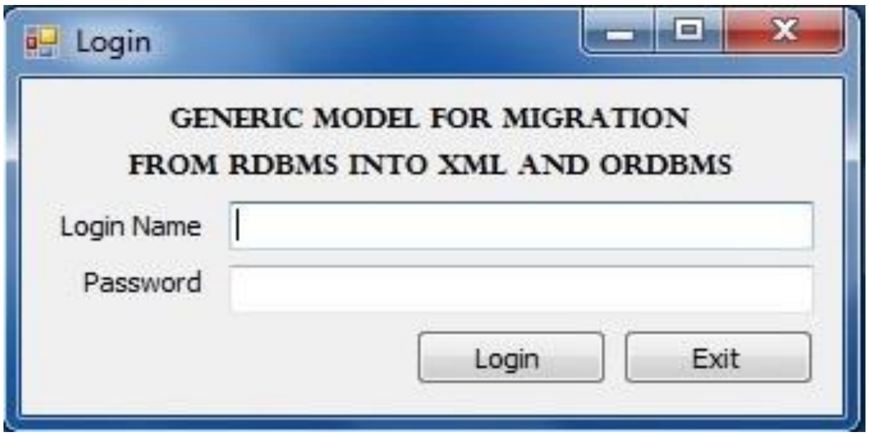


Fig. 5a. Login window.

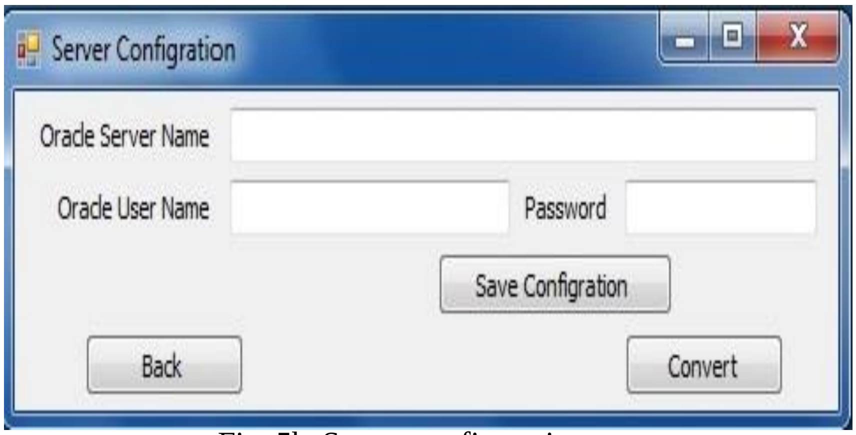
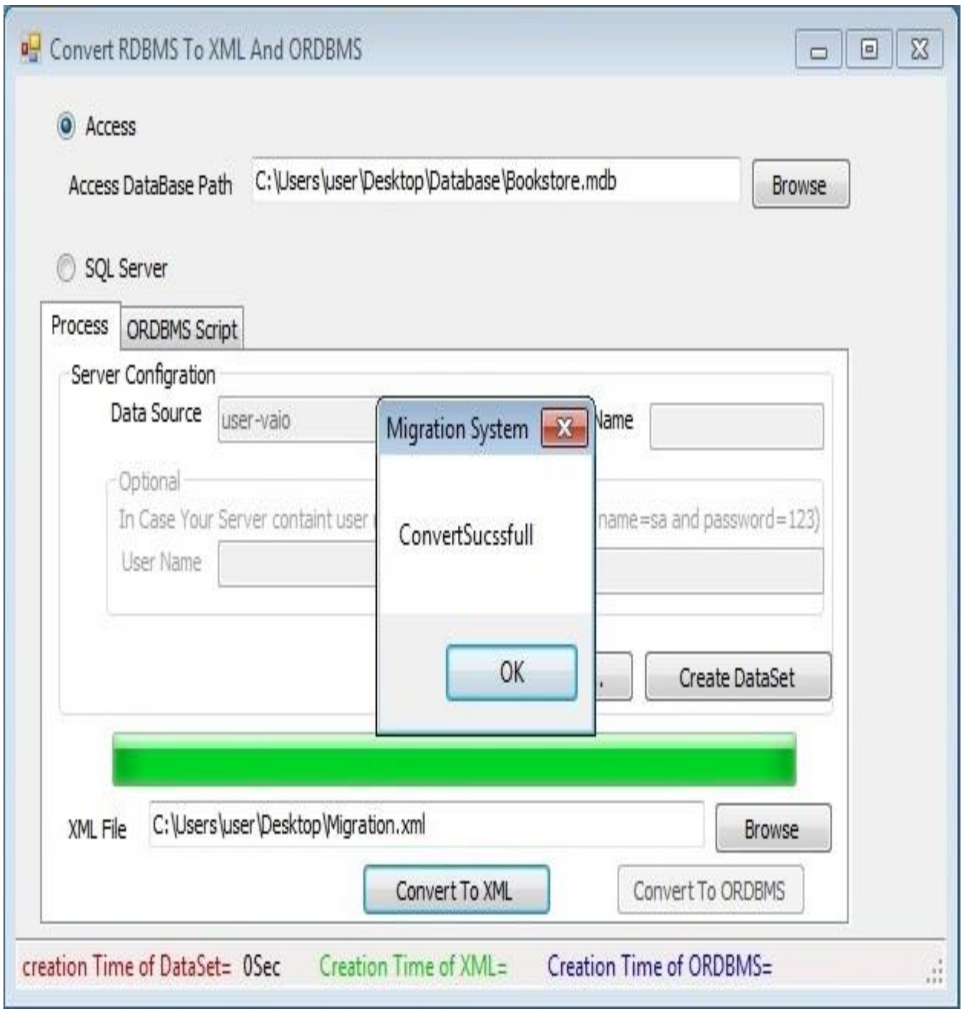


Fig. 5b. Server configuration.

The proposed algorithm is evaluated using two kinds of RDBMS such as SQL Server and ACCESS as an input. The result of conversion process as accepted on Orcale 11g as ORDB and XML document.

The system has been evaluated in terms of the equivalence of input and output databases. This has been performed by comparing the RDBs input into the system with its output; it was found that the output is equivalent with the input database. Also the algorithm's efficiency is gauged according to execution time of each task in the migration process. In this work, the algorithm has been tested by several examples. A sample example was randomly selected for discussion.

In this example, Access database has been chosen as an input for testing the algorithm. The first form is login window, as described above (Fig. 5a). The user must insert the login name and password and click to move to second form which asks the user to insert Oracle server configuration according to the Oracle user who needs to present the migrated tables on his/her account. Fig. 6 shows the Oracle server configuration used in this example. Then, "the conversion" buttons can be pressed to start the migration process.



Access DataBase Path: C:\Users\user\Desktop\Database\Bookstore.mdb

Data Source: user-vaio

In Case Your Server containt user … name=sa and password=123)

User Name

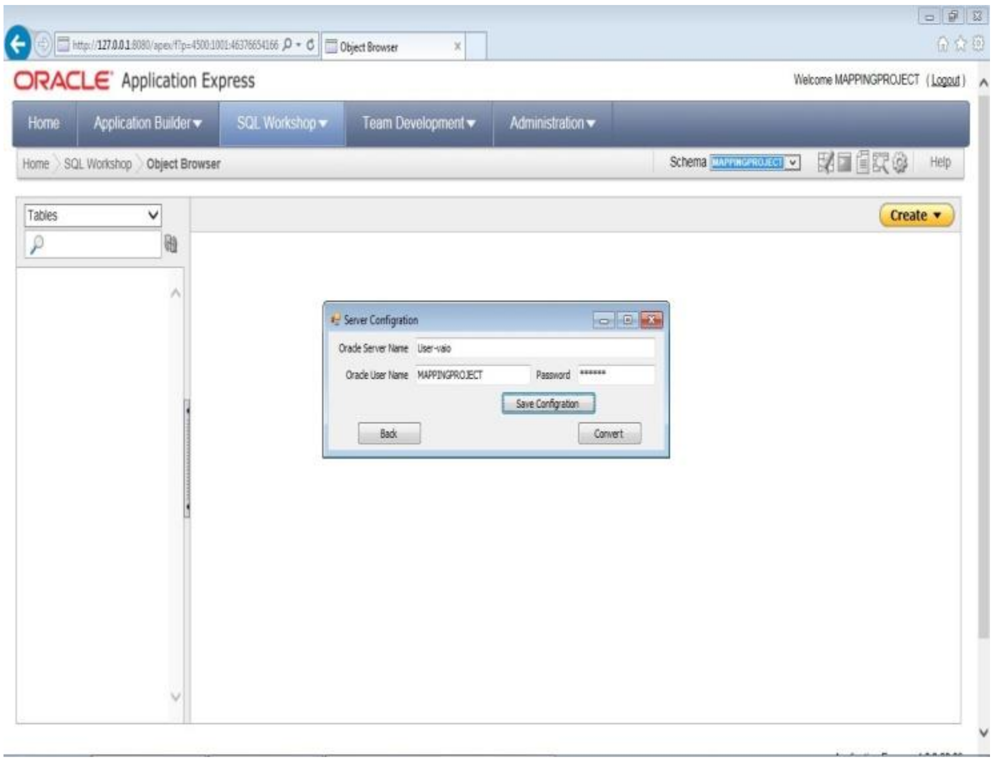
ConvertSucssfull

XML File: C:\Users\user\Desktop\Migration.xml

creation Time of DataSet= 0Sec Creation Time of XML= Creation Time of ORDBMS=

Fig. 5c. Main screen of the migration process.

As mentioned above, Fig. 5c shows the main screen of the migration process. It is required to enter the database path according to the chosen database. In our case, an Access database called "Bookstore" is inserted into the database path. This database consists of four tables. The "Test connection" button can be clicked to ensure that a successful connection is established. Then, the "Create Dataset" button enables the dataset creation. Finally, the "Convert to XML" can be pressed converting the input into an XML file. After converting into XML, two message boxes will be displayed. These are: "Relations added successfully" and "Data saved Successfully". Now, the



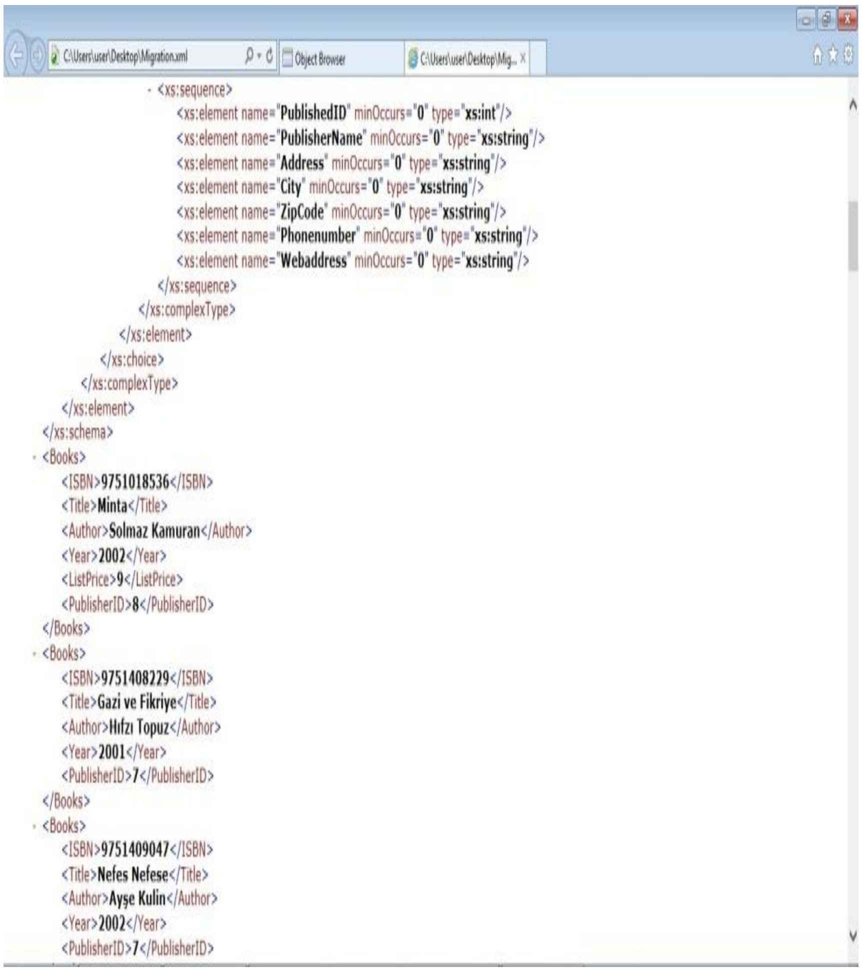
Oracle Server Name: User-vaio
Oracle User Name: MAPPINGPROJECT
Password: ******

Fig. 6. Oracle server configuration in our example.

migration process from RDBMS (Access) to XML is finished and the XML Browser is ready to explore the XML document (Fig. 7).

Optionally, clicking on "Convert To ORDBMS" button produces the same message boxes mentioned above. As a result, the migration process is done and the Oracle is ready to present the migrated database as shown in Fig. 8.

According to Fig. 9, it is noticeable that the execution times are calculated and displayed for each process. In our case, the creation time of the Dataset is 0 second, the creation time of an XML file is 4 seconds and the creation time of ORDBMS file is 4 seconds.



```
C:\Users\user\Desktop\Migration.xml        Object Browser        C:\Users\user\Desktop\Mig...
          - <xs:sequence>
                <xs:element name="PublishedID" minOccurs="0" type="xs:int"/>
                <xs:element name="PublisherName" minOccurs="0" type="xs:string"/>
                <xs:element name="Address" minOccurs="0" type="xs:string"/>
                <xs:element name="City" minOccurs="0" type="xs:string"/>
                <xs:element name="ZipCode" minOccurs="0" type="xs:string"/>
                <xs:element name="Phonenumber" minOccurs="0" type="xs:string"/>
                <xs:element name="Webaddress" minOccurs="0" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
- <Books>
    <ISBN>9751018536</ISBN>
    <Title>Minta</Title>
    <Author>Solmaz Kamuran</Author>
    <Year>2002</Year>
    <ListPrice>9</ListPrice>
    <PublisherID>8</PublisherID>
  </Books>
- <Books>
    <ISBN>9751408229</ISBN>
    <Title>Gazi ve Fikriye</Title>
    <Author>Hıfzı Topuz</Author>
    <Year>2001</Year>
    <PublisherID>7</PublisherID>
  </Books>
- <Books>
    <ISBN>9751409047</ISBN>
    <Title>Nefes Nefese</Title>
    <Author>Ayşe Kulin</Author>
    <Year>2002</Year>
    <PublisherID>7</PublisherID>
```

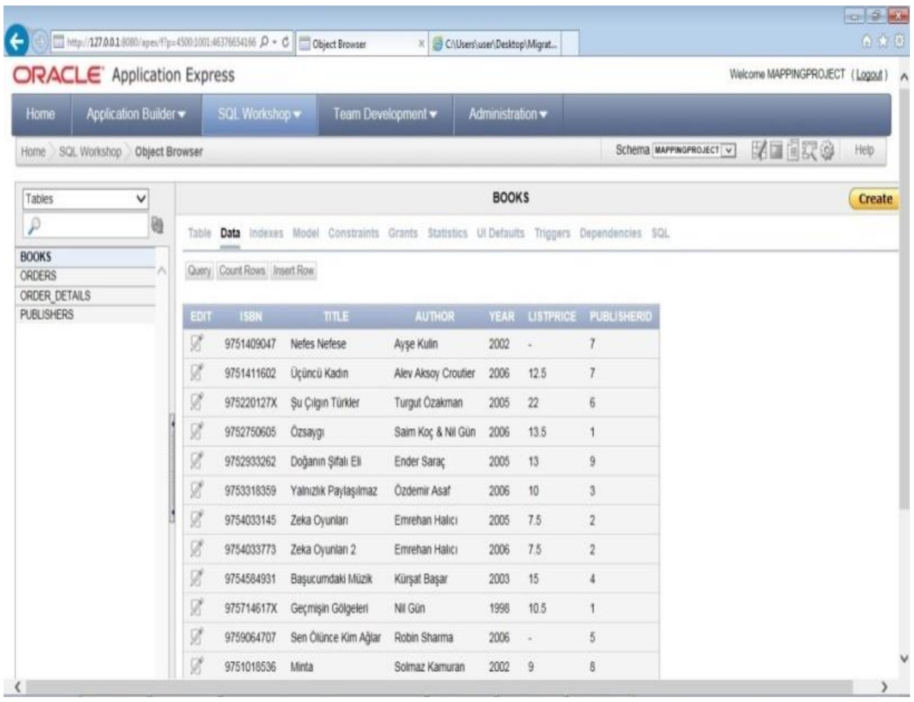
Fig. 7. An XML document.

CONCLUSIONS

This work proposes a migration algorithm which converts RDBMS into ORDBMS and XML data. The algorithm has been implemented using the Visual Basic .NET programming language for the sake of high developer productivity and easy maintainability. The implemented algorithm produces two different outputs. Data stored in RDBMS becomes available as ORDBMS and XML Data, which can be used by any other applications or systems even if it is running on another platform. Finally, the performance of the implemented migration algorithm is evaluated.

As a conclusion, it is clear that RDB can be automatically migrated into ORDB and XML data structure. The proposed algorithm migrated the RDBs into more than one target with preservation of the structure and semantics of an existing RDB and effectively converts the RDB data into the target databases without redundancy or loss of data.

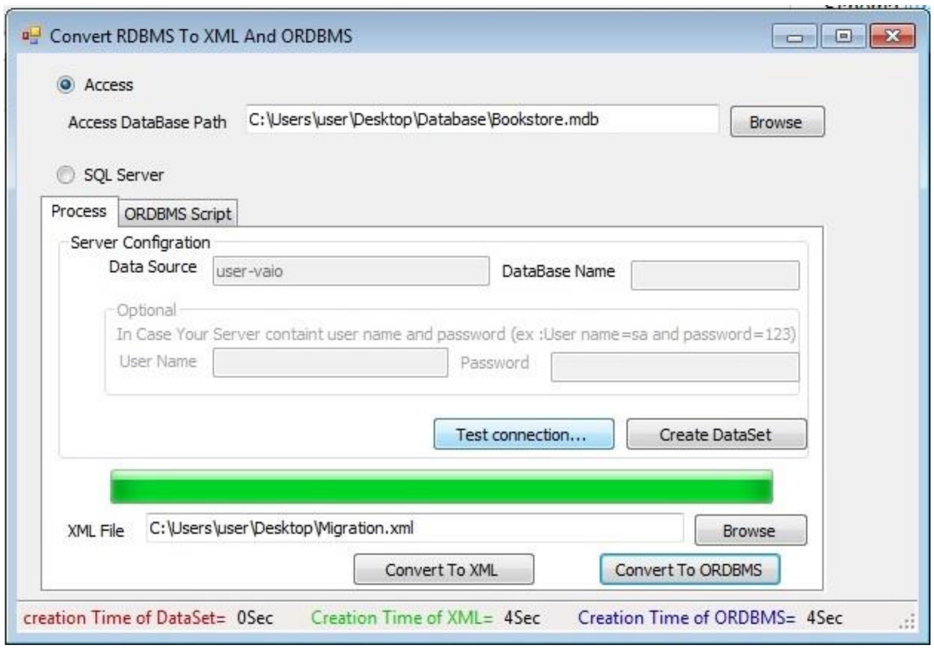
The implemented algorithm has been evaluated in terms of the equivalence of input and output databases. This has been performed by comparing the RDBs input of the



BOOKS

EDIT	ISBN	TITLE	AUTHOR	YEAR	LISTPRICE	PUBLISHERID
	9751409047	Nefes Nefese	Ayşe Kulin	2002	-	7
	9751411602	Üçüncü Kadın	Alev Aksoy Croutier	2006	12.5	7
	975220127X	Şu Çılgın Türkler	Turgut Özakman	2005	22	6
	9752750605	Özsaygı	Saim Koç & Nil Gün	2006	13.5	1
	9752933262	Doğanın Şifalı Eli	Ender Saraç	2005	13	9
	9753318359	Yalnızlık Paylaşılmaz	Özdemir Asaf	2006	10	3
	9754033145	Zeka Oyunları	Emrehan Halıcı	2005	7.5	2
	9754033773	Zeka Oyunları 2	Emrehan Halıcı	2006	7.5	2
	9754584931	Başucumdaki Müzik	Kürşat Başar	2003	15	4
	975714617X	Geçmişin Gölgeleri	Nil Gün	1998	10.5	1
	9759064707	Sen Ölünce Kim Ağlar	Robin Sharma	2006	-	5
	9751018536	Minta	Solmaz Kamuran	2002	9	8

Fig. 8. The ORDBMS on Oracle.



● Access

Access DataBase Path C:\Users\user\Desktop\Database\Bookstore.mdb

○ SQL Server

Server Configration

Data Source user-vaio DataBase Name

Optional

In Case Your Server containt user name and password (ex :User name=sa and password=123)

User Name Password

XML File C:\Users\user\Desktop\Migration.xml

creation Time of DataSet= 0Sec Creation Time of XML= 4Sec Creation Time of ORDBMS= 4Sec

Fig. 9. Main screen with execution times.

system with its output. It is observed that the output is equivalent to the input database. Also, the algorithm efficiency is evaluated according to the execution times of each task throughout the migration process.

REFERENCES

[1] Hert, M., Reif, G., and Gall, H. (2011). A comparison of RDB-to-RDF mapping languages. In Proceedings of the 7th InternationalConference on Semantic Systems, Graz, Austria, ACM, pp. 25-32.

[2] Elbekai, A. (2005). Mapping XML document to variety formats, 2nd ICETE International Conference on E-business Telecommunication Networks, UK, pp 9.

[3] Algusbi, E. (2010). Geniric Database Independent Application and XML based data exchange. Unpublished MSc Thesis, Academy of Graduate Studies, Tripoli.

[4] Albadwe, A. (2009). Mapping Object Oriiented Database into XML Data structure. Unpublished MSc Thesis, Academy of Graduate Studies, Tripoli.

[5] Monk, S., Mariani, J. A., Elgalal, B. and Campbell, H. (1996). Migration from rela-tional to object-oriented databases. Information & SoftwareTechnology, **38**(7), 467-475

[6] Wang, C., Lo, A., Alhajj, R. and Barker, K. (2004). Converting legacy relational database into XML database through reverse engineering, In ICEIS (1), p.216-221.

[7] Maatuk, A., Ali, A., & Rossiter, N. (2010). Converting relational databases into object relational databases. Journal of Object Technology, **9**(2), 145-161.

[8] Sumathi, S. and Esakkirajan, S. (2007). Fundamental of Relational Database Mangment Systems, Springer Verlag.

[9] "RDBMS & SQL Tutorial". Last accessed on May 2016, http://www.scribd.com/doc/6714724/Oracle-RDBMS-SQL-Tutorial

[10] Singh, S. K. "Database Systems: Concepts, Design and Applications", (2006). Pearson Education India. Safari books online, last accessed on November 2015. http://my.safaribooksonline.com/book/databases/9788177585674/relational-model/ch05

[11] "Weaknesses of RDBMS". Safari books online , last accessed on 10 November 2013, http://my.safaribooksonline.com/book/databases/9788177585674/object-relational-database/ch16lev1sec2

[12] "Object-relational DBMS". Safari books online , last accessed on 10 November 2013, http://my.safaribooksonline.com/9788177585674/ch16lev1sec1

[13] Farnoush Banaei-Kashani, (2008). "Introduction to ORDBMS". Last accessed on July 2015, http://infolab.usc.edu/csci585/Spring2008/Lectures/ORDBMS-Intro.pdf

[14] 'XML Essentials", W3C, (2010). Last accessed on January 2017, http://www.w3.org/standards/xml/core.html.

[15] "XML Basics", W3Schools. lLst accessed on March 2016 http://www.w3schools.com/xml/xml_usedfor.asp.

[16] "XML Elements", W3Schools. Last accessed on September 2015 http://www.w3schools.com/xml/xml_elements.asp.

[17] "XML Syntax", W3Schools. Last accessed on Novomber 2015 http://www.w3schools.com/xml/xml_syntax.asp.

[18] "XML Cdata", W3Schools. Last accessed on Feberaury 2016 http://www.w3schools.com/xml/xml_cdata.asp.

[19] "XML Advantages", ExforsysInc, 5th (2007). Last accessed on June 2016 http://www.exforsys.com/tutorials/xml/xml-advantages.html.

[20] "XML Schema", W3Schools. Last accessed on March 2017 http://www.w3schools.com/schema/schema_intro.asp.