

LINQ TECHNOLOGY USAGE WITH SQL, XML AND OBJECTS

¹BIRSEN G. OZDEMIR, ²EFTAL SEHIRLI, ³ANDAY DURU

¹Computer Engineering Department, Doğuş University, Turkey

^{2,3}Biomedical Engineering Department, Karabük University, Turkey

E-mail: ¹bgozdemir@dogus.edu.tr, ²eftalsehirli@karabuk.edu.tr, ³andayduru@karabuk.edu.tr

Abstract— LINQ which is the abbreviation of “Language Integrated Query” substantially provides the convenience to reach data from application side. Programmers can easily process database systems, xml documents and programming objects by using LINQ technology component. They can access programming object collections like datasets, arrays, lists by using LINQ to Objects component, database data by making SQL queries over tables as if they use database management systems thanks to LINQ to SQL component and xml tag values by querying over xml elements with LINQ to XML component. In this study, it is aimed to share the experience of LINQ technology usage in an education platform named as Graduation Project Management. This paper mainly focuses on specification of LINQ components utilization and main advantages of using LINQ to SQL in Graduation Project Management. It also brings differences between LINQ to SQL and Ado.Net to the fore and discusses why any future technology will be successful if it includes the main advantages of LINQ technology.

Index Terms— Ado.net, LINQ to SQL, LINQ to Objects, LINQ to XML.

I. INTRODUCTION

In the technology world, new products are introduced and also new properties of existing products are developed and added over old ones as time progresses. Some old technologies are not used anymore because new ones have more capabilities, work quicker and are used in a more simple way. LINQ has a power to fulfill these expectancies and will improve itself as it is used much more in different platforms including educational ones. Owing to launch with .Net 3.5, LINQ has started to be used by programmers. LINQ has some powerful properties distributed to different components such as LINQ to SQL, LINQ to XML and LINQ to Objects. This study predominantly expresses main characteristics of these three components of LINQ, shows usage methods of them and differences between Ado.Net and LINQ to SQL [1]. Finally, it concludes as ratio of LINQ usage will increase in education platforms if LINQ achieves to attract attention of programmers developing related applications. Moreover, instructors who intend to teach xml topic, database systems, and types of object collections like arrays, lists can prefer LINQ technology to reach their goals. Object oriented programming (OOP) can also be taught by the help of LINQ to SQL component to assist students for comprehension of related fundamental concepts in a rapid way [2].

II. METHODS

It has been aimed to bring to the fore main characteristics of three significant components in order that programmers who desire to benefit from advantages of LINQ can use it without any trouble, after making literature survey regarding LINQ technology. It has been seen that LINQ is more

frequently used with SQL, however it is not so LINQ to XML and LINQ to Objects components. In section 3.1, 3.2, and 3.3., you can find out the details. Formerly, commonly preferred component LINQ to SQL is used in Graduation Project Management (GPM) application and utilization results are shared in section 4 (Figure 1).

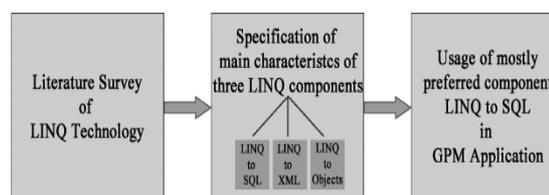


Fig.1. The flow of progress throughout this study.

III. GENERAL FEATURES OF LINQ

“Language Integrated Query” is unabbreviated form of LINQ. This technology was introduced to software world when .Net Framework 3.5 launched. LINQ can be used in different programming languages such as Java, JavaScript, PHP and .Net languages. LINQ offers easy code writing for programmers by making bridges between programming objects and data in different platforms. Thus, flow of information between objects and data is getting easier by using LINQ technology. While developing applications, developers can use different languages, platforms, technologies and so on. Nowadays, object oriented programming is regularly enabled by many of them. LINQ technology also supplies object oriented programming properties for developers to make them benefit from OOP advantages. Additionally, lambda expressions are available for LINQ technology. Thanks to them, making queries by using LINQ technology provides the convenient, easy and short code writing and

programmers get rid of creating new variables several times. Since lambda expressions can be in the form of any data type, a lot of goals can be fulfilled by writing fewer code statements.

A. Linq to Sql

LINQ to SQL is one of the components of LINQ technology that allows easily reaching data in Microsoft SQL Server databases from applications. Now, LINQ to SQL is used only for Microsoft SQL Server database systems. Nevertheless, there are some studies to work with LINQ for other database systems. LINQ technology assures easier utilization of object oriented programming benefits. That is to say that, this component creates each database table as a class in the application side. In addition to this, every field of tables in database side is generated as an attribute of related classes with the correspondingly correct data type. Creation of Report class based on the database table called `dbo.Report` is illustrated in Figure 2. When developers wish to reach and update data in database side, they can create instances/objects from related classes and then reach data from application side by using those instances/objects. Changes made on a database table by the help of created object from corresponding class are demonstrated in Figure 3.

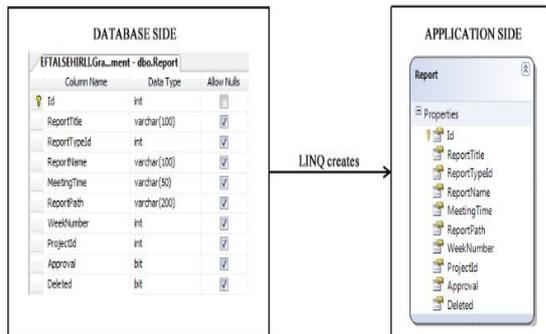


Fig.2. LINQ creates Report class based on the database table called `dbo.Report`.

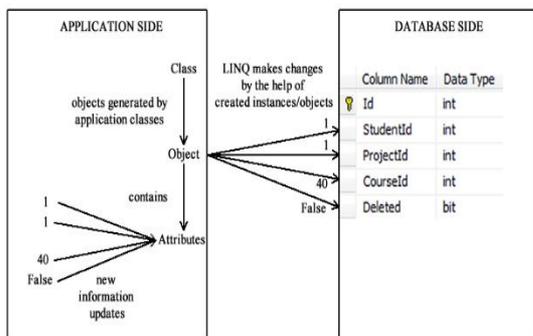


Fig.3. LINQ makes changes on a database table by the help of created object from corresponding class.

In application side, dynamic queries that are written in the format of LINQ by programmers have a little different format from SQL scripts in a database

system. However, there are also some similarities between both of them. For example, code statement below shows a standard LINQ to SQL syntax:

```
// LINQ to SQL code lines (standard)
public void GetStudent(int studentId)
{
try
{
GraduationProjectManagementDataContext db = new
GraduationProjectManagementDataContext();
var query = from s in db.Students
            where s.Id == studentId && s.Deleted == false
            select s;
gvStudentInformation.DataSource = query;
gvStudentInformation.DataBind();
}
catch (Exception)
{}}
```

This code is exactly equal to “select * from Students where Id=1” SQL script. According to LINQ code lines, `GraduationProjectManagementDataContext` is a class which is automatically produced by LINQ to SQL. `DataContext` word is added by LINQ to SQL component at the end of class name. Tables of the database can be reached if and only if an object created from this class exists. “s” is specified as the alias name for Student table. Subsequent line “select s” means that query returns all the attributes or fields of Student table. Result is held by query variable whose data type is var. Data returned from query are finally sent to gridview whose name is `gvStudentInformation`. As a result, information of student whose Id number is 1 is shown in the gridview. While writing LINQ to SQL queries, connection settings and database selection statements are not needed to be concerned by application developers.

There are also other LINQ query operators such as `GroupBy`, `OrderBy`, `Distinct`, `SelectMany`, `Join`, `Union`, `Intersect`, `Sum`, `Average`, `Single`, `Contains` and so on. Their usage follows similar logic with code statements stated in above example. It will be more illustrative for students to show these types of operators while teaching LINQ technology. In addition, these operators have same functionalities as if they exist in database side nevertheless, syntax rules are different in application side.

There are some advantages of LINQ to SQL such as preventing SQL injection, assuring OOP approach, using short code statements, providing type-safety, full-intellisense and rich refactoring properties. On the other hand, connection settings and database selection process are made while application side class is being created. LINQ to SQL automatically sets connection states like opening or closing at the background of the applications. Because of using object oriented approach, SQL injection, which means reaching a database or its tables to damage them by

writing SQL scripts from an application, is prevented. Even if SQL scripts like drop table are written in the application side, databases cannot directly be reached since they are carried as parameters thanks to LINQ objects [3]. Object oriented programming that is world widely known programming type has simplicity, reusability, modifiability and maintainability properties. Hence, LINQ also has these properties that increase the usage of LINQ to SQL. Simplicity makes software objects more understandable and code statements more readable by reducing complications. Reusability is provided by objects that can be reached from other applications and platforms. Modifiability offers that attributes or data types can be changed in programming side. Maintainability defends programming objects by encapsulating attributes and methods in LINQ classes. Type-safety property is another advantage that developers are able to write queries by seeing all options regarding attributes and methods of related classes (Figure 4). After putting dot sign, options are shown by LINQ technology. In addition, full-intellisense that stands for code completion is supplied for programmers (Figure 5). After writing a few letters and pressing space, tab and enter buttons together, options will be seen in a window. If there is only one option, auto code completion will be realized. Another advantage of LINQ technology is refactoring. It permits improving code lines by re-editing and shortening the length of them.

```
void GetAdvisorEmail() // get the list of advisors email
{
    try
    {
        GraduationProjectManagementDataContext db = new GraduationProjectManagementDataContext();
        var rs = from a in db.Advisors
                where a.Deleted == false
                select new { a.

```

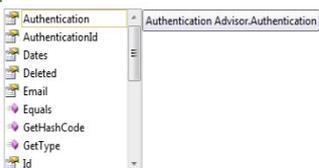


Fig.4. Type-safety property.

```
public void Save() // to save students' ids and their project ids
{
    try
    {
        GraduationProjectManagementDataContext db = new GraduationProjectManagementDataContext();
        StudentProject sp = new StudentProject(); // table instance is created
        sp.StudentId = this.StudentId; // column is filled
        sp.ProjectId = this.ProjectId; // column is filled
        sp.CourseId = this.CourseId; // column is filled
        sp.Deleted = this.Deleted; // column is filled
        db.StudentProjects.InsertOnSubmit(sp); // insert process
        db.Submit();
    }
    catch (Exception)
    {
        //
    }
}

```



Fig.5. Full-intellisense property.

LINQ to SQL has shorter code statements than Ado.Net. Because LINQ sets connections at the background and query result is directly transferred in a tool of application. However, Ado.Net cannot do same things. The example shown below expresses the difference between them clearly.

Ado.Net Code Sample:

```
public void GetStudentWithAdoNet(int number,
string name, string surname, int semester, double
GPA, bool Deleted)
{
    using (SqlConnection connection = new
SqlConnection(connectionString))
    {
        try
        {
            using (SqlCommand command = new
SqlCommand("Student", connection))
            {
                if (connection.State ==
ConnectionState.Closed)
                {
                    connection.Open();
                }

                command.CommandType =
CommandType.StoredProcedure;

                command.Parameters.AddWithValue("@StudentNu
number", number);
                command.Parameters.AddWithValue("@StudentNa
me", name);
                command.Parameters.AddWithValue("@StudentSur
name", surname);

                command.Parameters.AddWithValue("@StudentSe
mester", semester);
                command.Parameters.AddWithValue("@StudentGP
A", GPA);

                command.Parameters.AddWithValue("@Deleted",
Deleted);

                DataTable dt = new DataTable();
                SqlDataReader reader = command.ExecuteReader();
                while (reader.Read())
                {
                    dt.Load(reader);
                }
                gvStudent.DataSource = dt;
                gvStudent.DataBind();
            }
        }
        catch (Exception)
        {
        }
        finally
        {
            if (connection.State ==
ConnectionState.Open)
            {
                connection.Close();
            }
        }
    }
}
```

```

LINQ Code Sample:
public void GetStudentWithLinq(int number,string
name,string surname,int semester,double GPA,bool
Deleted)
{
    try
    {
        using
(GraduationProjectManagementDataContext db =
new GraduationProjectManagementDataContext())
        {
            var result = db.Students(number, name,
surname, semester, GPA, Deleted);
            gvStudent.DataSource = result;
            gvStudent.DataBind();
        }
    }
    catch (Exception)
    { }
}

```

These two code samples are doing same thing. But, LINQ to SQL has shorter code statements therefore it is quite readable and understandable better than ADO.NET, for which SqlConnection, SqlCommand, SqlDataReader, DataTable objects are created and put in the memory. The created objects and a while loop usage decreases the operating speed of application. On the other hand, LINQ to SQL use only one object and one variable, thus both do not cover too much spaces in the memory. Due to not having a loop, operating speed of application is not negatively influenced. Still, speed and performance are not significantly different from each other. Almost similar performance is generally obtained when using LINQ or ADO.NET code statements that have same functionalities.

B. Linq to Objects

Another component of LINQ technology is LINQ to Objects that use queries to obtain any result. It can utilize collections like array, ArrayList, List to make queries over them. When programmers desire to use array items which are matching with specified conditions, LINQ to Objects queries are very practical to reach the conclusion. If, if...else and switch clauses are not needed to be used in LINQ to Objects as they are not so effective due to returning the conclusion slower than the component. One example about this issue is shown below in a function called ListNumbersWithLinq. In this example, program output shows the numbers that are greater than 5.

```

public void ListNumbersWithLinq()
{
    int[] numbers = {0,1,2,3,4,5,6,7,8,9,10,11,12};
    var numbersList = from n in numbers
        where n>5
            select n;
}

```

```

foreach (var num in numbersList)
{
    Console.WriteLine(num);
}
}

```

Furthermore, LINQ queries provide three main advantages by using foreach loops:

1. They are quite short, understandable and readable, especially when multiple conditions are subjects.
 2. They provide powerful filtering, ordering, and grouping properties without writing lots of code statements.
 3. They can be reused in other data sources like GridView, ListBox without any changes over them.
- Instead of using iteration code statements for the mess operations, LINQ queries can overcome this issue very easily without causing any confusion. In the example above, fulfillment of the required condition is explained by using array. Another example using List data type is shown below.

```

public void StudentList()
{
    List<Student> StudentList =
Student.SampleData();
    var students = from student in StudentList
        where student.Deleted == false
            orderby student.Number ascending
            select student;

    gvStudents.DataSource = students;
    gvStudents.DataBind();
}

```

C. Linq to XML

XML which is abbreviation of Extensible Markup Language and used by both humans and computers is a language. To be able to be understandable language for the computers, some technologies must be used XML documents so as to read and write with some formatting rules. LINQ to XML is another used component of LINQ technology by creating queries over XML documents to realize to be understandable by computers. More than 50% time and 90% memory savings can be achieved by integrating LINQ with engineering applications to make the handling of XML-based data exchange more efficient [4].

Programmers can write queries in order to both read XML tag values and dynamically write new information if they want to put a constraint for the tag values. An example that explains clearly to make query over XML data is shown below. The following code statement selects all no, name and surname of people from loaded XML document. After making selection, data are stored in variable named people whose data type is var. Finally, GridView called gvPeople is filled with result of the query.

```

public void LoadXMLDocument(string path)
{
    try
    {
        XmlDocument xmlPeople =
        XmlDocument.Load(Server.MapPath(path));
        var people = from p in
        xmlPeople.Elements("people").Elements("person")
        select new
        {
            No = p.Element("no").Value,
            Name = p.Element("name").Value,
            Surname = p.Element("surname").Value
        };

        gvPeople.DataSource = people.AsEnumerable();
        gvPeople.DataBind();
    }
    catch (Exception)
    { }
}

```

A part of xml document that is loaded in the code statement above is shown below:

```

<?xml version="1.0" encoding="utf-8" ?>
<people>
  <person>
    <no>1</no>
    <name>Arda</name>
    <surname>Çelik</surname>
  </person>
  <person>
    <no>2</no>
    <name>Emin</name>
    <surname>Bal</surname>
  </person>
  <person>
    <no>3</no>
    <name>Erkut</name>
    <surname>Erkin</surname>
  </person>
  ...

```

Programmers have been introduced with var data type by launching of .Net 3.5. Even though var data type is thought to be analogous with var data type used in languages like Visual Basic or JavaScript, they are not equal to each other. In JavaScript or Visual Basic, var is used as a variant data type for the variables. In LINQ, however, var is not only a placeholder but also the actual data type, which is inferred by the context used in, is set at compile time (The Code Project). When programmers cannot decide what data type is used, var data type is a good option to get the data in correct forms.

Lambda expressions are one of the new features that are introduced with .Net 3.5. LINQ technology is the

one that utilizes these expressions, which are used for all data types in only c# programming language. Due to not having any constraint regarding data types, lambda expressions can check multiple data types in one code line and they provide easy code writing without declaring any new variables that must be suitable for values that will be checked. They are more readable and understandable. An example of lambda expressions is shown below to express them better:

```

// LINQ code lines with Lambda Expressions
GraduationProjectManagementDataContext db = new
GraduationProjectManagementDataContext();
Student student1 = db.Students.Select(p =>
p.Semester == 8 && p.GPA>3.50);

```

In this example, p is a lambda variable and => operator is lambda operator that defines p variable. In the example, two of attributes from student table are used. One of them is semester whose data type is integer and the second one is GPA whose data type is double. p lambda variable checks both integer variable and double variable in only one code line and result is returned as any data type. If students whose semesters are 8 and GPA are greater than 3.50 exist, student1 object will hold all information of students that are appropriate for these conditions.

IV. GPM APPLICATION

GPM is abbreviation of "Graduation Project Management" and designed for Doğuş University, İstanbul, Turkey. Advisors and students can influentially manage their graduation projects by using GPM that is a web application in the field of education. A view of GPM application for settings web page is shown in Figure 6. In this application, SQL Server 2008 R2 database management system is used in database side. While developing GPM, LINQ technology is preferred in order to interact with the database system. Students who took graduation projects from Computer Engineering Department in last years have been developed variant types of graduation project management applications for Doğuş University. However, LINQ technology has been used for the first time with GPM application. After specification and taking into account the requirements of customers, it is seen that LINQ to SQL component of LINQ technology was the most suitable component for GPM application. Other components such as LINQ to XML and LINQ to Objects were not needed to be used. Since XML documents were not wanted in order to get the information about students, advisors and projects. Thus, GPM did not include LINQ to XML usage. Furthermore, collections of data like arrayList, and list were not used in the application. For this reason, LINQ to Objects component was not used too. When

comparing GPM application with the old ones, in which LINQ to SQL were not preferred, it was realized that GPM is finished in shorter time. While developing the application, about two weeks were gained thanks to LINQ technology. GPM has less database code statements owing to usage of LINQ to SQL. This mainly explains why GPM application with LINQ to SQL was finished almost two weeks earlier than others [5].



Fig.6. A view of Settings popup menu of GPM Application.

CONCLUSION

LINQ is an efficient technology with its three components such as LINQ to SQL, LINQ to XML and LINQ to Objects that provide some advantages like interacting with database systems, xml documents and object collections from the application side. Instructors, who intend to teach database systems, xml topic and types of object collections like arrays, ArrayLists, and lists, can prefer LINQ technology to reach their goals. In addition, OOP can be taught by the help of LINQ to SQL component. Hence, students can comprehend related fundamental concepts in a quick way. For instance, most of IEEE members teach programming with LINQ technology by bringing to the fore an object first approach for students. They resume persist on teaching key components of LINQ framework as much as possible. They also teach how to benefit from LINQ to SQL for database programming [6]. This study reached a conclusion

regarding that LINQ to SQL has more predominant characteristics than Ado.Net such as short code length, code understandability and writing, doing refactoring easily, presenting object oriented programming property. Owing to valuable features of LINQ, it presents appropriate options in a lot of domains including educational ones. Object oriented programming, database systems, xml and some object collections are separately taught in universities or training areas. LINQ technology can be a big chance to easily teach all of them. On the other hand, as technology develops and provides amenities, people expect that any applications being web application, desktop application and mobile application ought to work in a faster manner that can be supplied by refactoring. Programmers while carrying out this kind of expectancies for applications that need refactoring can use technologies that provide speed, convenience, enhancement and short code length. In consequence, users will become satisfied with well worked applications. LINQ technology is one of the candidates that leads to succeed in the way of reaching this type of people's expectancies.

REFERENCES

- [1] Y. Wang and X. Zhang, "The research of multi-source heterogeneous data integration based on LINQ", International Conference on Computer Science and Electronics Engineering, pp. 147-150, 2012.
- [2] W. Y. Chung, P. S. Yu and C. J. Huang, "Cloud computing system based on wireless sensor network", Proceedings of the 2013 Federated conference on Computer Science and Information Systems, pp. 887-880, 2013.
- [3] A. Dave and S. Hate, "Building Tomorrow's Enterprise. Microsoft Data Access Technologies – SWT Analysis", Infosys, pp. 16-17, 2010.
- [4] M. Barth and A. Fay, "Efficient use of data exchange formats in engineering projects by means of language integrated queries – Engineers LINQ to XML", IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, Hamburg, pp. 1335-1340, 2010.
- [5] E. Şehirli and B. G. Özdemir, "Predominant characteristics of LINQ technology usage in education platforms", Cyprus International Conference on Educational Research (CY-ICER), 47, pp. 2084-2088, 2012.
- [6] W. Kan and Z. Yujun, "Using LINQ as an instructional bridge between object-oriented and database programming", Computer Science & Education ICCSE '09. 4th International Conference, pp. 1464-1468, 2009.

★★★