

# **STUDY AND ANALYSIS OF THE .NET PLATFORM- BASED TECHNOLOGIES FOR WORKING WITH THE DATABASES**

*Full paper*

**Irakli Basheleishvili<sup>1</sup>, Avtandil Bardavelidze<sup>2</sup>, Khatuna Bardavelidze<sup>3</sup>**

*<sup>1,2</sup>Department of Computer Technology, Akaki Tsereteli State University  
e-mails: basheleishvili.irakli@gmail.com*

*<sup>3</sup>Department of Automated Management Systems, Georgian Technical University  
Georgia*

**Abstract:** The paper deals with an overview of technologies for working with the relational databases on the Net Framework, and conducting a study for assessing the execution time of the data inserting, selecting, updating and deleting operations in a database by the ADO.NET, Dapper and Entity Framework technologies for working with the relational databases. An analysis was made based on the assessment results obtained.

**Key words:** ADO.NET, Dapper, Entity Framework, Relational databases.

## **1. INTRODUCTION**

The databases are widespread today, and they are used in virtually all branches of human activity. Almost every application has an information-based structure. The databases provide an effective, secure and flexible method of data storage, and they are used in various types of programs. In this context, similarly to a wide range of programming languages, there are many technologies to work with the databases that we use when developing software. Over the past few years, the non-relational databases have become relevant, but in spite of that, the relational databases do not lose their relevance, and they are still actively used today [1, 2].

Accordingly, the aim of our work is to introduce technologies to work with the relational databases (ADO.NET, Dapper, Entity Framework), which are supported

the .NET framework, and conduct a comparative analysis of them in terms of execution time of the Insert, Select, Update and Delete Operations.

There are numerous Internet postings, in which there is an attempt to compare the ADO.NET, Dapper and Entity Framework technologies with the databases, but no research paper is in this area that gives us a real picture of these technologies [3, 6, 15].

## **2. BASIC PART**

.NET Framework is a software platform developed by Microsoft. This platform is mostly used on operating systems of Microsoft Windows [5, 11]. The .NET Framework allows us for creating the desktop and web applications. Today, numerous applications are created on this platform on the software market, which is due to its multifunctionality and sophistication, which in turn is also due to the fact that it allows for programming in a variety of languages.

### **ADO.NET technology**

ADO.NET is part of the Microsoft .NET Framework library of basic classes. In fact, it is a library of object-oriented classes that allows for working with different data sources. The data source is the base of some relational data. The ADO.NET technology is integrated with the .NET Framework environment and is oriented to any .NET Framework language.

The set of classes integrated into the ADO.NET is divided into two groups: the first classes that store and manage data (for example, DataSet, DataTable, DataRow and DataRelation classes); the second classes that communicate with the databases (for example, Connection, Command and DataReader classes) [6, 7, 9, 14].

### **Dapper technology**

Dapper is an object-relational mapping (ORM) technology intended for the .NET Framework, which allows us for working with the database from programming language, particularly for executing operations in the program. It provides the framework for mapping an object-oriented domain model to a traditional relational database [15]. Dapper is free as an open source software, which is one of its strengths. Its main characteristics are as follows: speedy and high performance; the low number of software code lines; object mapper; multiple query support; support and easy handling of stored procedures; simplicity and flexibility of execution and parameters to the query; transaction support; ease of creating the objects by data obtained upon executing the operations; we are 100 per cent protected from the threat of sql injections when transferring parameters to the query.

### **Entity Framework technology**

Entity Framework is an object-relational mapper (ORM) technology that enables .NET developers to work with the databases using .NET framework objects. Its use

allows developers for avoiding the need for data-access code that they usually need to write [2, 4, 8]. If ADO.NET allows us for creating connections or for connecting other objects to the bases, Entity Framework is a higher level of abstraction that allows us for receiving information directly from the database and processing the data regardless their storage location, due to which it makes the programmer's work more effective, causing it to be widely used in large projects [10, 13].

### 3. PERFORMANCE EVALUATION

Execution time is one of the most important quality indicators for any software. While creating software, developers have to choose technologies that will minimize execution time of software. Considering that any software is based on a database, a major part of execution time is spent on the exchange of data between the program and database. It is therefore important for the information-based program to correctly select technology for working with a database, since the amount of execution time of the program largely depends on technology for working with a database.

Let us compare and analyze execution time a the Insert, Select, Update and Delete operations on a database by means of technologies for working with the databases, in programming language C# by the application specifically created for this purpose. The application allows us for conducting the experiment, executing four basic operations (insert, select, update and delete) on a database using all three technologies, and for determining the time required for their execution, as well as for undertaking their analysis. The application is related to a database created on a Ms Sql Server 2014. The experiment was conducted on a computer having the following parameters: Intel Core i5 2400s CPU 2.5 GHz, Operative Memory: DDR3 6GB, Operating System Windows 10 64bit.

#### Insert Operator

As is known, the Insert Operator allows us to enter (add) data to the database table. Let us conduct the experiment to determine how long it take ADO.NET, Dapper, Entity Framework technologies to execute the Insert Operator. The experiment involves adding both one and several records to the database table, simultaneously. The results of the experiment are given in Fig. 1, which clearly illustrates that using all three technologies on a database, the Insert Operator has been executed five. In the first time, there was made one database record, 10 records were made in the second time, 100 records - in the third time, 200 records - in the fourth time, 300 records - in the fifth time, and 400 records were made in the sixth time. Figure 1 illustrates a linear diagram showing the time used for the execution of the Insert Operator, according to technologies for working with the databases and the number of the database records.

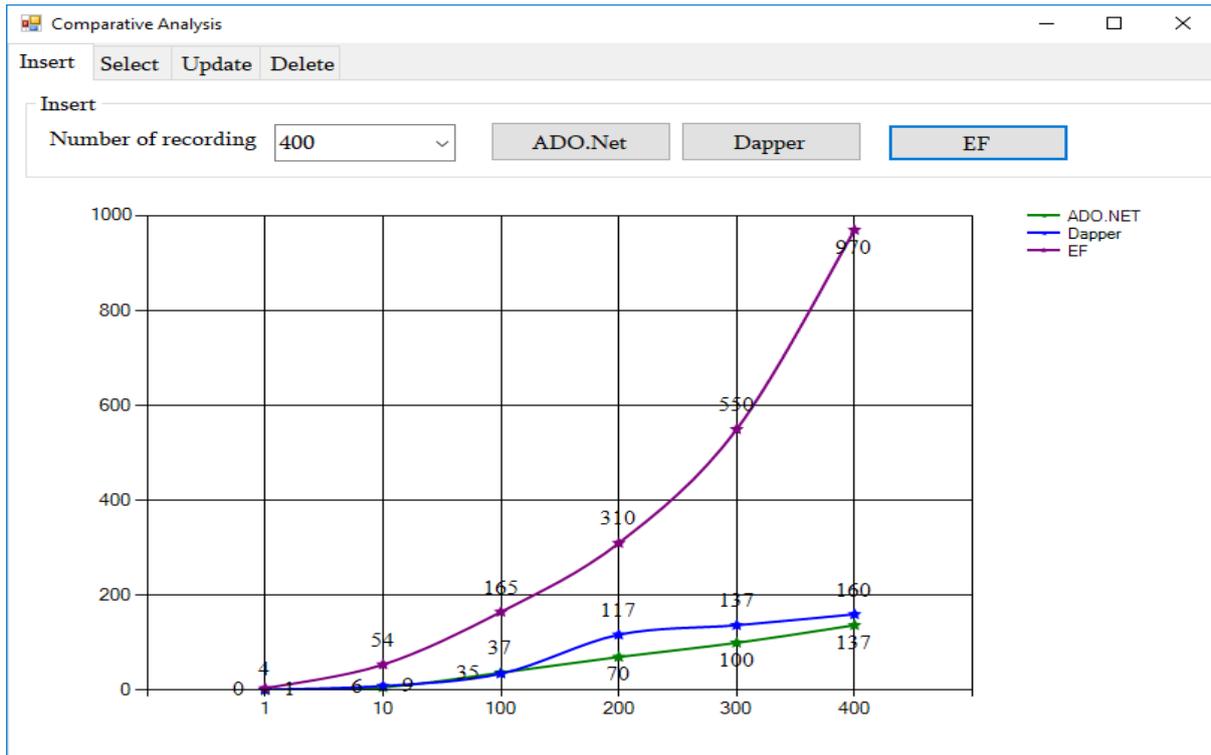


Fig. 1. Insert operator performance

On the linear diagram presented in Figure 1, along the Y-axis, we have a scale of times in milliseconds, and the number of the recorded data is given on the X-axis. A green color line on the diagram represents ADO.NET, a blue line is Dapper, while a lilac line represents Entity Framework. The linear diagram clearly shows the difference between the ADO.Net, Dapper and Entity Framework technologies according to the execution time for the Insert Operator. The Insert Operator's execution time by Entity Framework differs radically from ADO.Net and Dapper is radically different (higher), which increases together with the number of data in a database. As for the execution times of the ADO.Net and Dapper Insert Operator, they are the same (at minimum difference) at some stage, and at another some stage, there is a slight difference between them, particularly, the execution time with Dapper is higher in comparison with ADO.Net. According to this diagram, it is obvious that by the Insert Operator's execution times, ADO.NET is the fastest of these three technologies.

### Select Operator

Select Operator allows for reading the data from a database, which is also an important operation when working with the databases. Similar to the Insert Operator, the results of the experiment for Select Operator are presented in Figure 2.

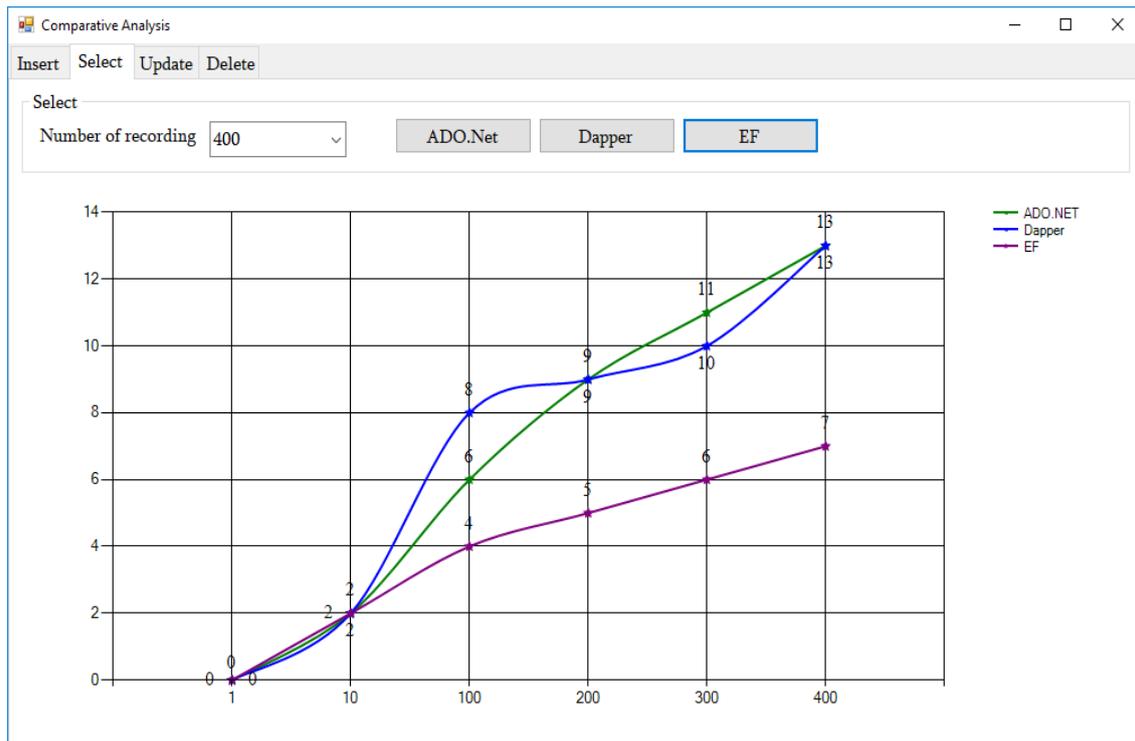


Fig. 2. Select operator performance

As illustrated in Figure 2, the linear diagram shows a large difference between the time of data reading and time of data writing, that is that data reading is executed faster than data writing. The difference is also expressed in case of data reading by using ADO.Net, Dapper and Entity Framework technologies, and a picture is totally different as compared with data writing that is reflected in the fact that data reading was executed most rapidly with Entity Framework. As for the ADO.Net and Dapper, we got a changing picture, in particular their time values were almost identical when reading 1 and 10 records from a database, while the reading time of 100 records with Dapper was higher than in case of using ADO.Net, and when reading 200 records, their execution time matched again. When reading 300 records, Dapper's time was lower as compared to ADO.Net, but when reading 400 records, their times became equals again.

### Update Operator

The results of the experiment conducted for assessing the data updating speed in a database by the ADO.Net, Dapper and Entity Framework technologies are presented in Figure 3, which illustrates the linear diagram almost identical to the diagram that we obtained in case of Insert Operator. Accordingly, ADO.Net is the fastest technology, while Entity Framework is the slowest technology.

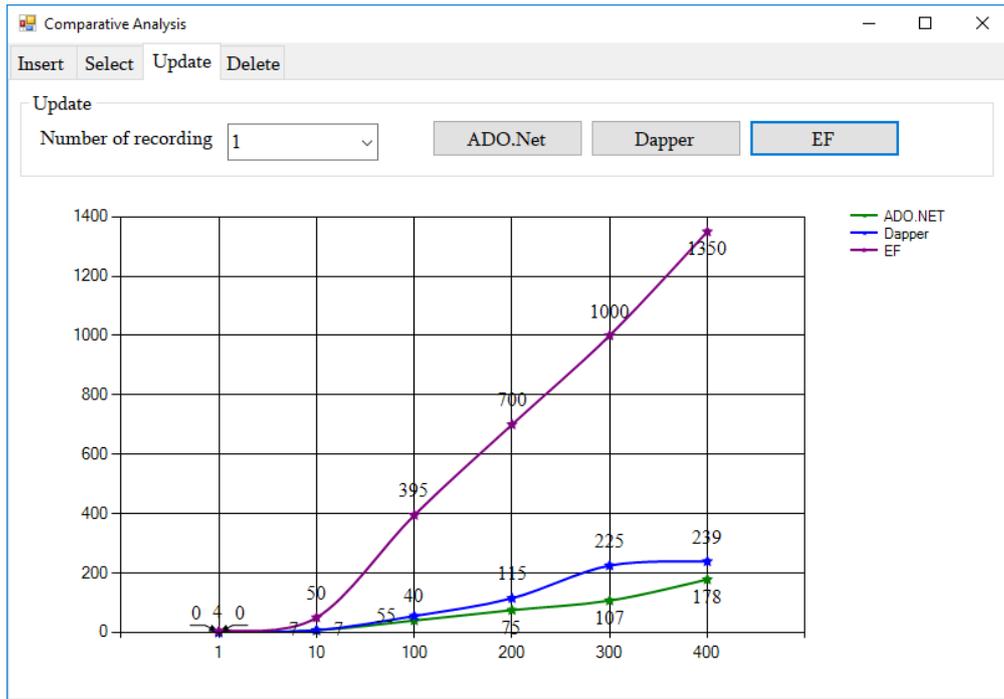


Fig. 3. Update operator performance

### Delete operator

The result of the data delete experiment is shown in Figure 4, which illustrates the linear diagram showing that Entity Framework is the slowest technology executed the delete operation, while the data deleting speeds of ADO.Net and Dapper technologies are almost identical, with a slight difference.

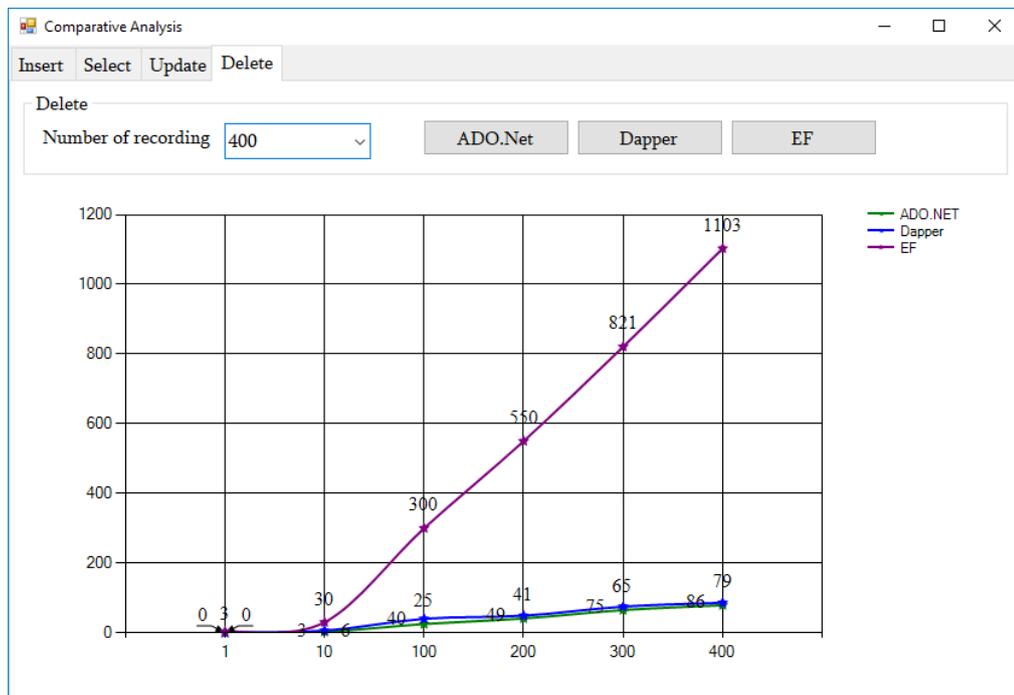


Fig. 4. Delete operator performance

## 4. CONCLUSION

In this paper, we present technologies for working with the databases of the modern and popular software platform, as well as with the well-known databases, such as ADO.NET, Dapper and Entity Framework. We discussed their main characteristics and capabilities. We conducted a study on execution time of the Insert, Select, Update and Delete operations in a database by the presented technologies. To that end, we developed a simple application in programming language C#. The results of this study revealed that ADO.NET technology was the fastest in the execution of the Insert, Update and Delete operations, and Entity Framework technology was the slowest (with a radical difference in time), as for Dapper technology, its result was not radically different from ADO.NET technology. In order to ensure high accuracy of the study results presented in the paper, we conducted experiments ten times (the paper describes one experiment), as a result of which, we calculated the average indicators, by which we got the same picture as in the results of one experiment presented in this paper.

## REFERENCES

- [1] Zhou, X., & Zhuo, Z. H. (2014). Research and Realization of ADO. NET Database Access Technology. *In Applied Mechanics and Materials* Vol. 496, pp. 1748-1751.
- [2] Wiphusitphunpol, W., & Lertrusdachakul, T. (2017, June). Fetch performance comparison of object relational mapper in .NET platform. In 2017 14th International Conference on Electrical Engineering/Electronics, *Computer, Telecommunications and Information Technology* (ECTI-CON) (pp. 423-426). IEEE.
- [3] Lendvai, A. J., & Shi, H. (2007). ADO and ADO .NET Object Model Comparisons: A Relational Perspective. *IJCSNS*, 7(1), 330.
- [4] Blakeley, J. A., Muralidhar, S., & Nori, A. (2006). The ado. net entity framework: Making the conceptual level real. In *International Conference on Conceptual Modeling* (pp. 552-565). Springer, Berlin, Heidelberg.
- [5] Hibbert, A. (2003). ADO. Net Examples and Best Practices for C# Programmers. *ITNOW*, 45(1), 30-30.
- [6] HUA, G. D., & LIU, W. Y. (2004). Performance Optimizing in Database Accessing Based on ADO .NET [J]. *Application Research of Computers*, 6.
- [7] Wildermuth, S. (2003). *Pragmatic ADO. NET: Data Access for the Internet World*. Addison-Wesley Professional.
- [8] Lerman, J., & Miller, R. (2011). *Programming Entity Framework: Code First: Creating and Configuring Data Models from Your Classes*. " O'Reilly Media, Inc."
- [9] Vaikunth Pai, T., & Aithal, P. S. (2017). Disconnected Data Access Architecture using ADO .NET Framework. *International Journal of Applied Engineering and Management Letters* (IJAEML), 1(2), 10-16.
- [10] Lerman, J. (2010). *Programming Entity Framework: Building Data Centric Apps with the ADO. NET Entity Framework*. " O'Reilly Media, Inc."
- [11] .NET Framework (available at: <https://docs.microsoft.com/en-us/dotnet/>)
- [12] Dapper (available at: <https://dapper-tutorial.net/dapper>)

- [13] Entity Framework (available at: <https://docs.microsoft.com/en-us/ef/>)
- [14] ADO.NET (available at: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview>)
- [15] Dapper vs Entity Framework vs ADO.NET Performance Benchmarking (available at: <https://exceptionnotfound.net/dapper-vs-entity-framework-vs-ado-net-performance-benchmarking/>)