



African Journal of Advanced Pure and Applied Sciences (AJAPAS)

Online ISSN: 2957-644X

Volume 3, Issue 1, January-March 2024, Page No: 11-15

Website: <https://aaasjournals.com/index.php/ajapas/index>

معامل التأثير العربي 2023: (1.55) SJIFactor 2023: 5.689 ISI 2022-2023: 0.557

Comparative Analysis of ADO.NET and Entity Framework Performance Evaluation in Database Operations

Abdullah Abdalsalam Abdasameea^{1*}, Tahr Salem Daduch², Husayn Mansour Isbeeqaq³,
Hadeeyah Mohammed Shakeeb⁴

^{1,2,3,4} Computer Department, Faculty of Education, Alasmarya Islamic University, Zliten, Libya

*Corresponding author: a.abdulsami@asmarya.edu.ly

Received: October 30, 2023

Accepted: December 27, 2023

Published: January 10, 2024

Abstract:

This paper aims to compare the performance of ADO.NET and Entity Framework in various database operations, including insert, update, delete, and retrieve. The study investigates the execution times of these operations and analyzes the findings to determine the efficiency of each technology.

The research findings reveal that ADO.NET consistently outperforms Entity Framework in terms of execution time for insert, update, and delete operations. ADO.NET demonstrates faster execution times, indicating its effectiveness and efficiency in handling these database tasks. However, in the retrieve operation, Entity Framework unexpectedly demonstrates superior performance, suggesting its efficacy in retrieving data.

These results emphasize the importance of considering the specific requirements and performance characteristics of ADO.NET and Entity Framework when choosing the appropriate technology for different database tasks. Organizations and developers need to weigh the advantages and disadvantages of each technology based on their specific needs.

Further research and analysis are necessary to delve deeper into the underlying factors contributing to the performance differences observed in this study. This research provides valuable insights into the performance comparison of ADO.NET and Entity Framework and serves as a foundation for future investigations in this area, facilitating informed decision-making in database technology selection.

Keywords: Ado.net, Entity Framework, Database, Performance

Cite this article as: A. A. Abdasameea, T. S. Daduch, H. M. Isbeeqaq, H. M. Shakeeb, "Comparative Analysis of ADO.NET and Entity Framework Performance Evaluation in Database Operations," *African Journal of Advanced Pure and Applied Sciences (AJAPAS)*, vol. 3, no. 1, pp. 11–15, January-March 2024.

Publisher's Note: African Academy of Advanced Studies – AAAS stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2023 by the authors. Licensee African Journal of Advanced Pure and Applied Sciences (AJAPAS), Libya. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

مقارنه تحليلية بين (ADO.NET) و (Entity Framework) لتقييم الأداء في عمليات قواعد البيانات

عبدالله عبدالسلام عبدالسميع^{1*}، طاهر سالم دعدوش²، حسين منصور اسبيقه³، هدية محمد اشكيب⁴
^{1,2,3,4} قسم الحاسوب، كلية التربية، الجامعة الأسمرية الإسلامية، زلiten، ليبيا

الملخص

تهدف هذه الورقة البحثية إلى مقارنة أداء تقنيتي ADO.NET و Entity Framework في عمليات قاعدة البيانات المختلفة، بما في ذلك الإدراج والتحديث والحذف والاسترجاع للبيانات. تبحث الدراسة في سرعة اختلاف تنفيذ هذه العمليات وتحليل النتائج لتحديد كفاءة كل تقنية.

تشير نتائج البحث بتفوق ADO.NET كل مرة على Entity Framework من حيث سرعة التنفيذ لعمليات الإدراج والتحديث والحذف، مما يشير إلى فعاليته وكفاءته في التعامل مع مهام قواعد البيانات. غير أن Entity Framework تتفوق في عملية استرجاع البيانات بشكل غير متوقع، مما يشير إلى فعاليته في استرداد البيانات. لا تقوم هذه الورقة بتفضيل تقنية على أخرى بل يتم تحديدها من خلال نوع العمليات السائدة في النظام المستهدف انشأؤه. وعلى المطورين مراعات الموازنة بين مزايا وعيوب كل تقنية بناءً على احتياجاتهم الخاصة. يوفر هذا البحث رؤى قيمة حول مقارنة أداء ADO.NET و Entity Framework ويعمل كأساس للتحقيقات المستقبلية في هذا المجال، مما يسهل اتخاذ قرارات مستنيرة في اختيار تقنيات الاتصال بقواعد البيانات.

الكلمات المفتاحية: ADO.NET، Entity Framework، قاعدة البيانات، الأداء.

Introduction

In terms of development aspects, every information communication technology application or system is built on the fundamental and underlying activities of access, transit, storage, and data manipulation. One of the data-processing technologies must be chosen in order to enable database linking of the application during development. ADO.NET and Entity Framework are two of the most important Microsoft technologies for data access in software development. [1]

ADO.NET and Entity Framework are two of Microsoft's Object-Relational Mapping (ORM) frameworks that enable .NET developers to work with relational database systems. While the two frameworks have some overlap in their capabilities

The ADO.NET framework provides a low-level API to interact with databases through queries and stored procedures. It requires developers to handle most of the complexities of object-oriented paradigms like SQL query generations, caching, and change tracking. The Entity Framework, on the other hand, provides a higher level of abstraction with an object-oriented paradigm. It handles many ORM complexities under the hood and exposes entities and relationships in code. The Entity Framework's level of abstraction allows for the rapid development of database-driven applications. [1]

It is essential for software developers who work with data to understand the differences and similarities between these two technologies. This paper aims to assist developers in making knowledgeable decisions when choosing a data access technology for their projects by offering a comparative analysis of ado.net and entity framework.

The approaches for including databases in the project are discussed.

The main tasks for a data-oriented application are CRUD operations (creating, reading, updating, and deleting) [2]. Both Entity Framework and ADO.NET are in charge of these activities. Code execution speed was measured with these fundamental operations.

Material and methods

1. Ado.net

ADO.NET is a data access technology that is used to interact with databases in the Microsoft .NET Framework. It provides a consistent interface for accessing data from different types of data sources, including SQL Server, Oracle, MySQL, and others. ADO.NET is designed to provide high-performance data access, with support for both connected and disconnected data access scenarios. [3][4]

- **Connected Architecture:** The Connected Architecture of ADO.NET refers to a scenario where the application maintains a continuous connection to the database. In this architecture, the application opens a connection to the database, executes SQL commands against the database, and then closes the connection. This architecture is suitable for scenarios where the application needs to maintain a continuous connection to the database, and the data is accessed frequently. [1][4]
- **Disconnected Architecture:** The Disconnected Architecture of ADO.NET refers to a scenario where the application connects to the database, retrieves data from the database, and then disconnects from the database. In this architecture, the application retrieves data from the database and stores it locally in a dataset or data table. The application can then manipulate the data locally without requiring a continuous connection to the database. This architecture is suitable for scenarios where the data is accessed infrequently or where the application needs to work with data offline. [1][4]

In this paper, as far as ADO.NET is concerned, the connected architecture was used for the development of the application

2. The Entity Framework

Entity Framework is an Object Relational Mapper (ORM) developed by Microsoft, which enables developers to work with relational data using business models. It eliminates the need for developers to write low-level data access code with ADO.NET by providing a complete, model-based system. The framework has evolved

significantly since its first release, Entity Framework 3.5, which was included in .NET Framework 3.5 SP1 and Visual Studio 2008 SP1. With the current version being 6.0, Entity Framework simplifies the creation of a data access layer by providing an abstraction layer that removes the need for writing repetitive data access code for each domain model. [1][6][7]

Entity Framework facilitates the creation of a data access layer by allowing data access and describing the data as a conceptual model, that is, a set of entities and relationships. The application can handle basic CRUD (create, read, update, and delete) actions as well as one-to-one, one-to-many, and many-to-many relationships between entities. [1][5]

Entity Framework supports three primary approaches to building the Entity Data Model (EDM).

- The Database First approach is used when working with an existing database schema. In this approach, the EDM is generated from the database schema, which provides a convenient way to build applications that rely on an existing database. [1][5]
- On the other hand, the Code First approach involves writing domain models as classes, which form the basis of the EDM. The database schema is then generated from these models. This approach is ideal for applications that are highly domain-centric and where the domain models are created first, with the database serving as a persistence mechanism. [1][5]
- Finally, the Model First approach is similar to Code First, but with the added benefit of a visual EDM designer. The EDM is visually designed by using this tool, and the database schema and classes are generated based on the conceptual model. This approach also provides SQL statements needed to create the database, which can be used to connect the application to the database.[1][5]

In this paper, as far as Database First is concerned, the Database First approach was used for the development of the application

3. Application

The application was used in the College of Education Alsmarya Islamic University. It was created with Visual Studio 2015 with C# programming language as an integrated development environment (IDE) and Microsoft SQL Server Management Studio 2017 for the database. The application's goal is to make it easier to work at the College of Education by creating a study plan that takes into account the lack of conflict in the halls. The following are the application's features:

- Access to the system is granted through the department heads, and each department head is only authorized to interact with data pertaining to his department.
- Faculty member's data can be added, viewed, modified, and deleted by the Head of Department.
- The system administrator is granted the following rights:
 - Payment for new users
 - See the complete department schedule.
 - Add, change, or remove classroom data. Add, update, or delete classroom data using hall data. Course information
- The department head is the only person who may add, show, alter, or delete lectures from the academic schedule.
- When adding a lecture, it is confirmed that it is not being used from another department at the same time
- Print the academic timetable by department
- Print the academic schedule for the college
- Determine the hall's vacant times using its number.

Information on the device used in the comparison

OS Name	Microsoft Windows 10 Pro
Version	10.0.19045 Build 19045
Processor	Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz, 2201 Mhz, 2 Core(s), 4 Logical Processor(s)
BIOS Version/Date	Dell Inc. A07, 10/16/2015
Installed Physical Memory (RAM)	4.00 GB

4. Diagram for database

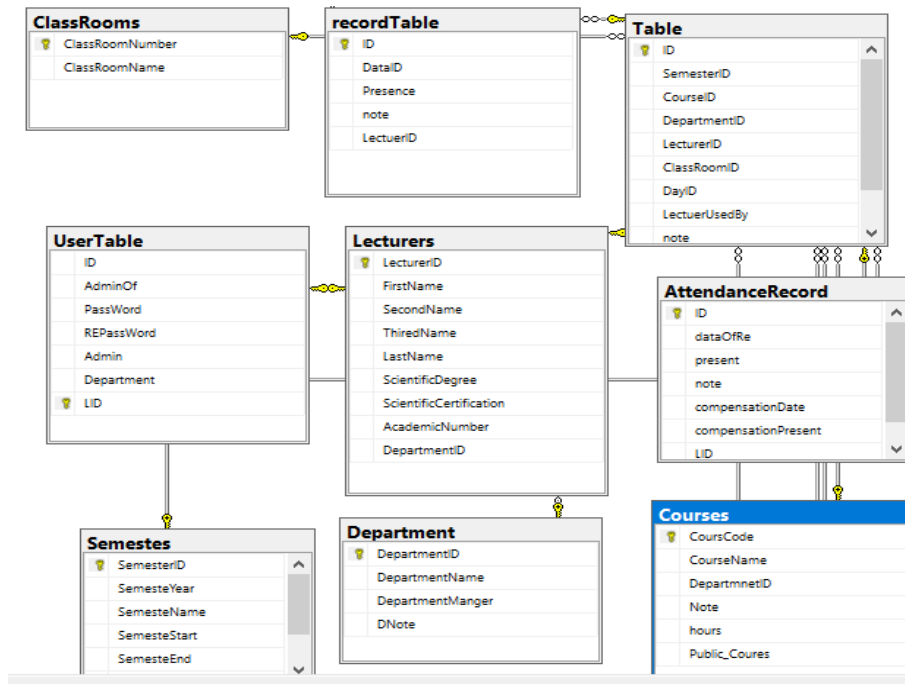


Figure 1: Diagram for database.

5. IDE Microsoft Visual Studio

Microsoft's Visual Studio is an integrated development environment (IDE). It includes a comprehensive collection of software development tools and functions such as code editing, debugging, testing, and deployment. Visual Studio supports a wide range of programming languages, including C#, C++, Python, JavaScript, and many more, making it a powerful tool for a wide range of development tasks.

Developers can use Visual Studio to efficiently write, edit, and manage code. It has powerful code editing tools including syntax highlighting, code completion, and intelligent code suggestions that assist enhance productivity and reduce errors. The IDE also provides debugging tools that allow developers to find and solve bugs in their code while it is being developed. [8]

The Methodological Framework of The Research

The comparison was conducted by measuring the execution speed of various operations, including insert, update, delete, and retrieve, using both ADO.NET and Entity Framework. The lecturers' table was used as an example.

Note that the table has a relationship with the following tables (users, department, and academic schedule)

The execution times in milliseconds were converted to seconds and recorded for each operation, and the average execution speed was calculated. All operations.

No	Entity Framework				Ado.net			
	insert	update	delete	retrieve	insert	update	delete	retrieve
1	0.005	0.006	0.007	0.010	0.002	0.004	0.006	0.011
2	0.004	0.003	0.004	0.008	0.002	0.003	0.003	0.010
3	0.005	0.006	0.005	0.009	0.001	0.001	0.001	0.010
4	0.085	0.011	0.004	0.005	0.001	0.004	0.004	0.015
5	0.011	0.050	0.011	0.007	0.002	0.003	0.001	0.010
6	0.006	0.006	0.005	0.006	0.001	0.001	0.001	0.011
7	0.004	0.004	0.003	0.012	0.003	0.003	0.003	0.009
8	0.006	0.006	0.005	0.008	0.003	0.004	0.001	0.008
9	0.004	0.005	0.003	0.008	0.002	0.003	0.002	0.010
10	0.006	0.012	0.003	0.005	0.001	0.001	0.001	0.008
AVG	0.136	0.109	0.05	0.078	0.018	0.027	0.023	0.102

Results and discussion

The research findings reveal interesting insights into the performance comparison between Entity Framework and ADO.NET for various database operations. The measurements conducted in this study provide valuable data on the average execution times of insert, update, delete, and retrieve operations for both technologies.

When analyzing the insert operation, it is evident that ADO.NET achieved an impressive execution time of 0.018 seconds, while Entity Framework took an average of 0.136 seconds. This highlights the superior performance of ADO.NET in terms of database inserts.

Moving on to the update operation, Entity Framework recorded an average execution time of 0.109 seconds, whereas ADO.NET exhibited better efficiency with an average time of 0.027 seconds. This demonstrates that ADO.NET excels as a data updater, providing faster execution for update operations.

Regarding the delete operation, ADO.NET showcased faster execution times with an average of 0.023 seconds, whereas Entity Framework took 0.05 seconds on average. These results reaffirm the performance advantage of ADO.NET in database deletions.

Interestingly, the retrieve operation showcased a different outcome. Despite ADO.NET's overall superiority in other operations, Entity Framework displayed slightly better performance in retrieval tasks. However, the difference in execution times between the two technologies was not significant.

Conclusion

In conclusion, the research findings indicate that ADO.NET outperforms Entity Framework in terms of execution time for the insert, update, and delete operations. ADO.NET consistently exhibited faster execution times compared to Entity Framework, showcasing its efficiency in handling these database operations. However, in the retrieve operation, Entity Framework unexpectedly demonstrated superior performance with its average execution time, suggesting its effectiveness in retrieving data. These results highlight the importance of considering the specific requirements and performance characteristics of each technology when choosing between ADO.NET and Entity Framework for different database tasks. Further research and analysis are necessary to delve deeper into the factors influencing the performance differences observed in this study.

References

- [1] ILIĆ, Miloš, et al. "The difference between ado. net and entity framework in software development." *Annals of the University of Oradea: Fascicle of Management & Technological Engineering* 2 (2021).
- [2] Blakeley, J. A., Muralidhar, S., & Nori, A. V. (2006). *The ADO.NET Entity Framework: making the conceptual level real*. In *Lecture Notes in Computer Science* (pp. 552–565). https://doi.org/10.1007/11901181_42
- [3] Mcleblanc (no date) *Overview - ADO.NET*, *Overview - ADO.NET* | Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview> (Accessed: 06 December 2023).
- [4] *Dot Net Tutorials About the Author: Pranaya Rout*Pranaya Rout has published more than 3 (2023) *Introduction to ADO.NET*, *Dot Net Tutorials*. Available at: <https://dotnettutorials.net/lesson/what-is-ado-net/> (Accessed: 06 December 2023).
- [5] Mcleblanc (no date a) *Architecture - ADO.NET*, *Architecture - ADO.NET* | Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-architecture> (Accessed: 01 December 2023).
- [6] Sheta , A. M., et al. (2017). *A Comparative Study between ADO.NET and Entity Framework*. *International Journal of Engineering Trends and Technology*, 43(4), 204-209.
- [7] Mcleblanc. (2021a, September 15). *Entity Framework Overview - ADO.NET*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>
- [8] Anandmeg (no date) *What is Visual Studio?*, Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022> (Accessed: 01 December 2023).