مجلة جامعة بني وليد للعلوم الإنسانية والتطبيقية
Bani Waleed University Journal of Humanities and Applied Sciences
تصدر عن– جامعة بني وليد – ليبيا
Website:https://jhas-bwu.com/index.php/bwjhas/index

المجلد العاشر – العدد الرابع – 2025   الصفحات (34- 50)   ISSN3005-3900

Evaluating ADO.NET Technology for Efficient Database Management: Concepts, Models, and Comparative Analysis

Hisham Ibrahim El Tajouri [1]*, Abdulhakim Moawa Dreheeb [2]
[1] Computer Department, Higher Institute of Science and Technology-Alshmokh Tripoli- Tripoli, Libya
[2] Computer Department, Faculty of Humanities and Applied Sciences, Al-Zaytouna University, Tarhuna, Libya
hishamtaj052@gmail.com

تقييم تقنية ADO.NET لإدارة قواعد البيانات بكفاءة: المفاهيم، النماذج، والتحليل المقارن

هشام إبراهيم التاجوري [1] *، عبد الحكيم معاوية دريهيب [2]
[1] قسم الحاسوب، المعهد العالي للعلوم والتكنولوجيا- الشموخ طرابلس، طرابلس، ليبيا
[2] قسم الحاسوب، كلية العلوم الإنسانية والتطبيقية ،جامعة الزيتونة، ترهونة، ليبيا

تاريخ الاستلام: 2025-08-17   تاريخ القبول: 2025-09-09   تاريخ النشر: 2025-10-01

الملخص:
تناولت هذه الدراسة تقنية ADO.NET بشكل عام، من حيث بنيتها الأساسية وأنواع الاتصال التي توفرها. كما قارنت بين هذه التقنية وتقنيات أخرى لإدارة قواعد البيانات مثل ORM، إضافة إلى مقارنتها بالإصدار الأقدم منها وهو ADO التقليدية .
ولتحقيق أهداف البحث، تم استخدام المنهج الوصفي التحليلي لشرح الجوانب النظرية، إلى جانب المنهج المسحي لجمع البيانات من عينة مكونة من ثلاثين طالبًا بقسم تقنيات الحاسب في المعهد العالي للعلوم والتقنية الشموخ.
أظهرت الدراسة أن ADO.NET يمثل حلًا عمليًا وفعّالًا لسوق العمل الليبي، خاصةً للمؤسسات التي تواجه تحديات مثل ضعف استقرار الإنترنت، التعامل مع قواعد بيانات ضخمة، أو الحاجة لتطوير مهارات عملية للمطورين المحليين. تتيح المعمارية الهجينة معالجة البيانات محليًا قبل المزامنة، مما يقلل الاعتماد على الاتصال المستمر ويزيد كفاءة التطبيقات الكبيرة والمعقدة، كما يسهل نموذج الاتصال المنفصل استخدام التقنية في بيئات الاتصال المحدودة.
وأظهرت نتائج الاستبيان أن غالبية الطلبة يجدون ADO.NET أسهل في التعلم وأفضل أداءً مقارنة بـ ADO التقليدية وORM، ويفضلون استخدامها في مشاريعهم العملية. لذلك،
توصي الدراسة باستخدام ADO.NET في المشاريع التي تتطلب تحكمًا عاليًا وكفاءة في الأداء، دمجها ضمن المناهج الأكاديمية، وتعزيز التدريب العملي للطلبة والمطورين لضمان الاستخدام الاحترافي. كما يُنصح بتطبيق أفضل الممارسات لمعالجة مشاكل الاتصال وتحسين الأداء العام، والاستمرار في تقييم التقنية وتحديث مهارات المستخدمين لضمان استدامة الفائدة وتحقيق أعلى مستويات الكفاءة المؤسسية.

الكلمات الدالة: قاعدة البيانات، ADO.NET، نموذج البيانات المتصل، نموذج البيانات المنفصل، Data Access، NET Framework، بنية ADO.NET ، بيانات.XML.

Abstract
The study highlighted that ADO.NET offers a practical and reliable solution for the Libyan labor market, particularly for organizations dealing with unstable internet, large datasets, or the need to strengthen practical skills among local developers. Its hybrid architecture allows data to be processed locally before synchronization, reducing reliance on constant

connectivity and improving the performance of large and complex applications. The disconnected connection model makes it especially useful in environments with limited or inconsistent network access.
Survey results showed that most students found ADO.NET easier to learn and more efficient compared to traditional ADO and ORM, and they preferred using it in their own projects. Based on these findings, the study recommends adopting **ADO.NET for projects that demand high control and efficiency**, integrating it into academic curricula, and providing hands-on training to ensure developers and students can use it effectively. Organizations are also encouraged to implement best practices for managing connectivity challenges and to continuously update skills and assess the technology to maximize its benefits and maintain high levels of operational effectiveness..

Keywords: Database, ADO.NET, Connected Data Model, Disconnected Data Model, Data Access, NET Framework, ADO.NET Architecture, XML data.

Introduction

In the digital age, data is now seen as an essential resource used by businesses of all sizes. The dramatic rise in data volume has led to an increasing need for sophisticated data management systems that guarantee the precision, security, and efficiency of data processing. To handle internal data, such as employee records, and external data, such as website analytics and customer profiles, it's critical to have a strong and adaptable connection framework with databases.

One of the most well-known technologies that allows programs to interact effectively with databases is the ADO. NET technology. It is a component of the . NET Framework and is notable for its capacity to interface with a variety of data sources, such as XML files, Oracle, SQL Server, and Microsoft Access. Additionally, ADO. NET fully supports two connection models: connected and disconnected. This gives application developers a great deal of freedom when it comes to designing applications, optimizing performance, and using fewer resources.[1]

The wellstructured programming model of ADO. NET makes it easy to carry out commands, retrieve data, and manage data effectively. It makes use of essential elements like DataAdapter, DataSetand Connection controls, which enable data manipulation independent of its origin and facilitate database updates and synchronization.

In contrast to the previous ADO technology, which relied on a persistent database connection throughout operations, ADO. NET uses a disconnected paradigm where data can be processed locally in memory and the connection is only restored when modifications are necessary. Because it helps to increase scalability, reduce network load, and enhance application speed, this method is especially well-suited for web applications and services that need to handle vast amounts of data at once.

This Study examines the architecture and underlying principles of ADO. NET, contrasts connected and disconnected data models, and analyzes its performance in comparison to older technologies like traditional ADO and numerous ORMs. To create a solid theoretical foundation, address the study questions, and offer an unbiased evaluation of the use of this technology, the study employs a descriptive analytical and survey approach.

2. Study Problem and Questions:

Managing database activities often presents difficulties for web and software developers since older technologies like ADO which lacked flexibility and support for offline data management frequently cause issues. Microsoft created ADO. NET in answer to these issues; it is a superior and more flexible tool for accessing and managing data across several platforms. Furthermore challenging is choosing the right data access technology, which strikes a compromise among performance, maintainability, and interface with other application modules. Although Object Relational Mapping (ORM) systems provide abstraction, they can also bring overhead or restrict the degree of control accessible. This study aims at showing how ADO behaves. NET provides developers a lot of database transaction control, hence perfect for programs emphasizing security and speed.[2]

In this Study , we are looking for answers to these questions:
What is ADO.NET technology?
What is the difference between ADO.NET and its previous version (ADO)?
What is ADO.NET Architecture?
What are the core components of ADO.NET and what role does each play in data management?
What is the difference between connected mode and disconnected mode in ADO.NET?
What are the advantages (features) of ADO.NET technology?
What is the difference between ADO.NET and other data access systems, such as ORM, and traditional ADO?"

Goals of the study:

to assess students' knowledge of ADO.NET, traditional ADO, and ORM.
to determine the primary features and challenges of ADO.NET and to observe how simple it is for students to learn.
to find out how students feel about ADO.NET's functionality, particularly when it comes to large data and complex applications.
to evaluate ADO.NET's usefulness in relation to alternative data access techniques.
to make recommendations on how to incorporate ADO.NET into school courses and enhance hands-on training.

Significance of the Study :

This **Study** seeks to assist developers and IT professionals in effectively managing data using ADO.NET. By understanding its architecture and capabilities, developers can build more efficient applications that connect to, retrieve, and manipulate data securely and reliably.

Study methodology:

Descriptive Analytical Method: Aims to gather sufficient information about the theoretical framework of the study so that, by reviewing relevant literature and previous Study on ADO.NET technology, the study questions can be answered.

Descriptive Survey Method: Used to collect field data on the usage of ADO.NET technology compared to other systems. This method was applied to a sample of 30 students enrolled at the Higher Institute of Science and Technology AlShomoukh in Tripoli, which is affiliated with the Ministry of Technical and Vocational Education. The results will be presented in the second chapter of the study.

6. Previous Studies:

Past studies have investigated the evolution of data access technologies, particularly the shift from ADO to ADO.NET, focusing on how performance and scalability have improved.

The following previous studies are relevant to this study:

Sivakumar and Arulprakash (2018) conducted a comparative study between ADO and ADO.NET, focusing on the architectural differences and data handling capabilities. Their Study concluded that ADO.NET offers superior scalability and is more suitable for modern distributed applications due to its disconnected data model and XML integration.[1]

Al-Salman & Al-Debei (2015) analyzed data access strategies in .NET environments, highlighting how ADO.NET supports multi-tier architecture and reduces server load by leveraging the in-memory dataset model. They recommended ADO.NET for enterprise-scale solutions where performance and reliability are critical.[3]

Gupta (2012) explored how ADO.NET can be utilized in real-time database applications. The study demonstrated that the use of ADO.NET reduces the complexity of database operations and simplifies transaction management, especially when using DataAdapter and DataSet classes for disconnected data processing.[4]

Ahmed &Majeed (2020) investigated the use of ADO.NET in educational management systems and concluded that the technology provides an efficient way to manage student data and course records while minimizing connection overhead and enhancing data security through parameterized queries.[5]

These studies support the view that ADO.NET represents a significant improvement over earlier data access technologies and plays a crucial role in optimizing database interaction, particularly in distributed and web-based applications.

7. Section One: Theoretical Framework on ADO.NET

7.1 Concept of ADO.NET:

ADO.NET is a collection of classes that come with the .NET Framework and allow us to retrieve data using the languages supported by this framework.

ADO.NET enables us to access relational data, such as databases like Microsoft Access and SQL Server, as well as other data sources. It can also be used to access non-relational data sources.

The ADO.NET classes are found inside an assembly called **System.Data.dll**, which contains code libraries named **System.Data** and **System.Xml**. ADO.NET supports working with XML data.[6]

Why is it called ADO.NET?

The name comes from an older technology called ActiveX Data Objects, or ADO. This technology was a set of classes used in older programming languages like Visual Basic 6 to access databases. Microsoft chose the name ADO.NET to indicate that this technology is the new and improved way to access data in the .NET environment.

ADO.NET performs the same functions as ADO, but it is more modern, easier to use, and more organized through object-oriented programming.

Consequently, developers can create apps that are both potent and simple to use, thanks to the increased flexibility and speed of data management provided by ADO. NET.

7.2. ADO. NET vs. ADO differences and the history of ADO. NET

ADO, developed in the 1990s, was limited by its reliance on constant connections and restricted XML features even though it provided a basic data access model. The 2002 release of ADO. NET of the. NET Framework addressed these shortcomings by providing disconnected access and easy XML integration.[7]

The switch from ADO to ADO. NET answered the growing requirements of web and business application development by providing a completely managed environment with improved error handling, better type safety, and greater support for disconnected operations replacing the COM components and synchronous operations ADO depended on.

This shows Microsoft's deliberate transition in data access models toward more scalability, modularity, and enhanced security.

The following table illustrates the key differences between ADO and ADO.NET

Feature	**ADO**	**ADO.NET**
Architecture	Connected (Continuous connection)	Disconnected (Intermittent connection)
Data Representation	Recordset	DataSet / DataTable
XML Support	Limited	Extensive
Performance	Slower in web applications	Optimized for .NET applications
Scalability	Limited scalability	High scalability
Integration	COM-based	Native to .NET Framework

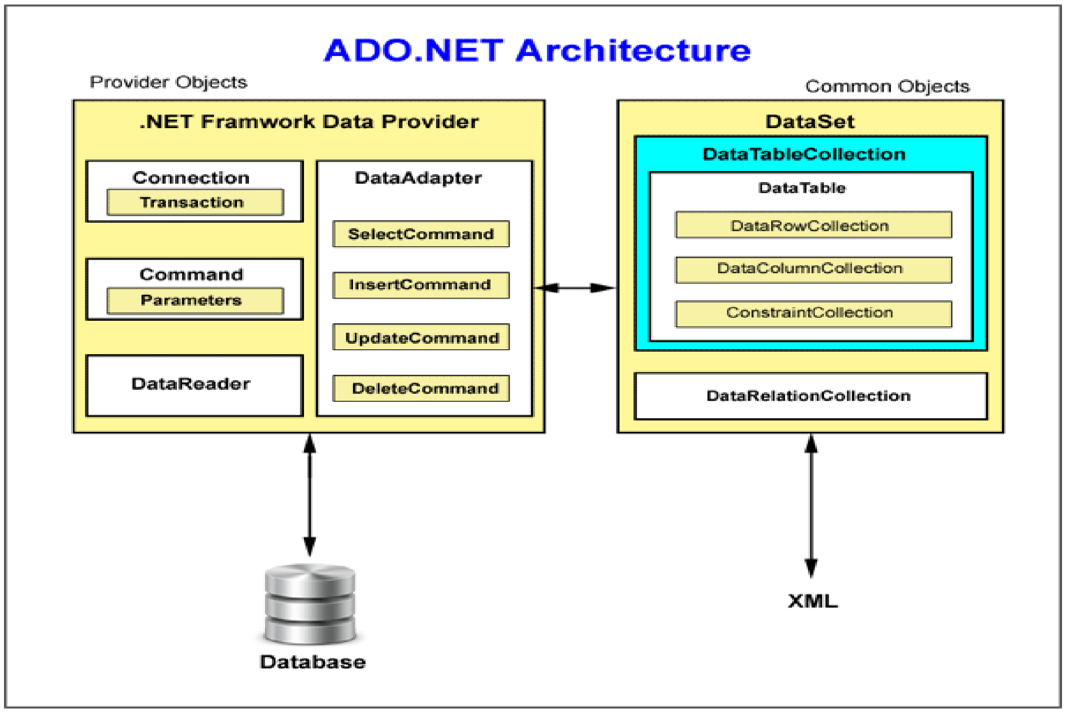
Table 1: Key Differences Between ADO and ADO.NET

7.3. ADO.NET Architecture:

Designed to help consumers more efficiently get, administer, and store data than other data access technologies, ADO. NET Architecture It is meant to manage connected as well as disconnected data access scenarios.

The interplay of elements like Connection, Command, DataReader, DataAdapter, and DataSet creates a flexible architecture. This modular architecture lets data operations span several platforms—cloud, web, mobile, and desktop. Furthermore, the architecture is easily matched with XML Web Services, hence allowing ADO. NET to operate efficiently in service oriented application contexts.[8]

Let's see how the ADO.NET components comprise the architecture



The figure (1) represents the ADO.NET components model and how they work together[11]

7.4. Connected and Disconnected Data Models:

For the duration of data analysis, the connected model's application maintains a continuous link with the database. Though it might not function well for larger user groups, it helps in realtime systems where instant changes are needed.

The disconnected model, by contrast, enables programs to engage with a local copy of the data, modify it offline, and then sync the changes back to the database. For mobile applications and dispersed systems when a constant internet connection is not always present, this is especially useful.

Not only does the disconnected approach boost scalability but it also enables mobile and offline-first applications, in which constant connectivity cannot be guaranteed. Before sending changes back to the database upon reconnection, developers using ADO. NET's DataSet and DataTable can carry out complex clientside operations including sorting, filtering, and data binding. [5]

The connected and disconnected modes are summarized as follows:

Connected Mode:

Throughout the course of operations, the connection to the database remains open, providing quick and easy access to the data.

Disconnected Mode:

Changes are then transmitted back after the connection is restored; data is fetched into memory and handled locally without the need for a continuous database connection.[9]

The table below shows the primary contrasts between the Connected Model and the Disconnected Model.

Aspect	**Connected Model**	**Disconnected Model**
Connection Status	Continuous	Temporary
Complexity	Simpler	More complex
Scalability	Limited	High
Usage	Small, simple apps	Multi-tier, distributed apps

Table2: Comparison between Connected Model and Disconnected Mode

7.5. ADO.NET Objects:

Data access technique of the . NET Framework, ADO. NET, is used for obtaining or modifying data, executing queries, and opening database connections. It includes objects meant for usage with both connected and unconnected data access techniques.

Table below shows the ADO. NET components and their associated methods. [6]

ADO.NET Object	**Function / Description**
Connection	Establishes database link
Command	Executes queries or stored procedures
DataReader	Reads data in forward-only mode
DataAdapter	Moves data between DB and DataSet
DataSet	Holds in-memory data
DataTable	Represents a single table

DataRow/DataColumn	Row or column in a DataTable
DataRelation	Links multiple tables
CommandBuilder	Auto-generates SQL for updates
Parameter	Enables safe parameter passing
Connection	Establishes database link
Command	Executes queries or stored procedures

Table 3: Main ADO.NET Objects and Functions

7.6. Advantages of using ADO.NET:

Simple Data Access: With the help of ADO. NET, data may be easily accessed from a variety of data sources, including databases and XML files. ADO. NET provides a collection of classes that allow developers to easily handle data.[12]

High Performance: Since it lowers the strain on the database, data architecture that uses ADO. NET to disconnect is simple to expand. Since all processing takes place on the client side, performance is improved.

Because ADO. NET uses disconnected data access, the application does not hold the database connections for an extended period of time, making it scalable. Because of this, numerous users may access the data, making it scalable.

Maintainability: In ADO. NET, the data logic is distinct from the user interface. This enables us to build our program in distinct layers. As a result, it's simple to maintain.[13]

Active Community: A lot of developers are actively using ADO. NET, and there is a ton of community help for it. This makes it simpler for a novice who is just starting out to learn how to use it. There is also a lot of documentation available.

In conclusion, ADO. NET is a robust and adaptable data access technology for . NET applications, with a feature set that makes it a great option for creating reliable and effective applications.

Section Two: Research Methodology and Procedures

8.1 Methodology:

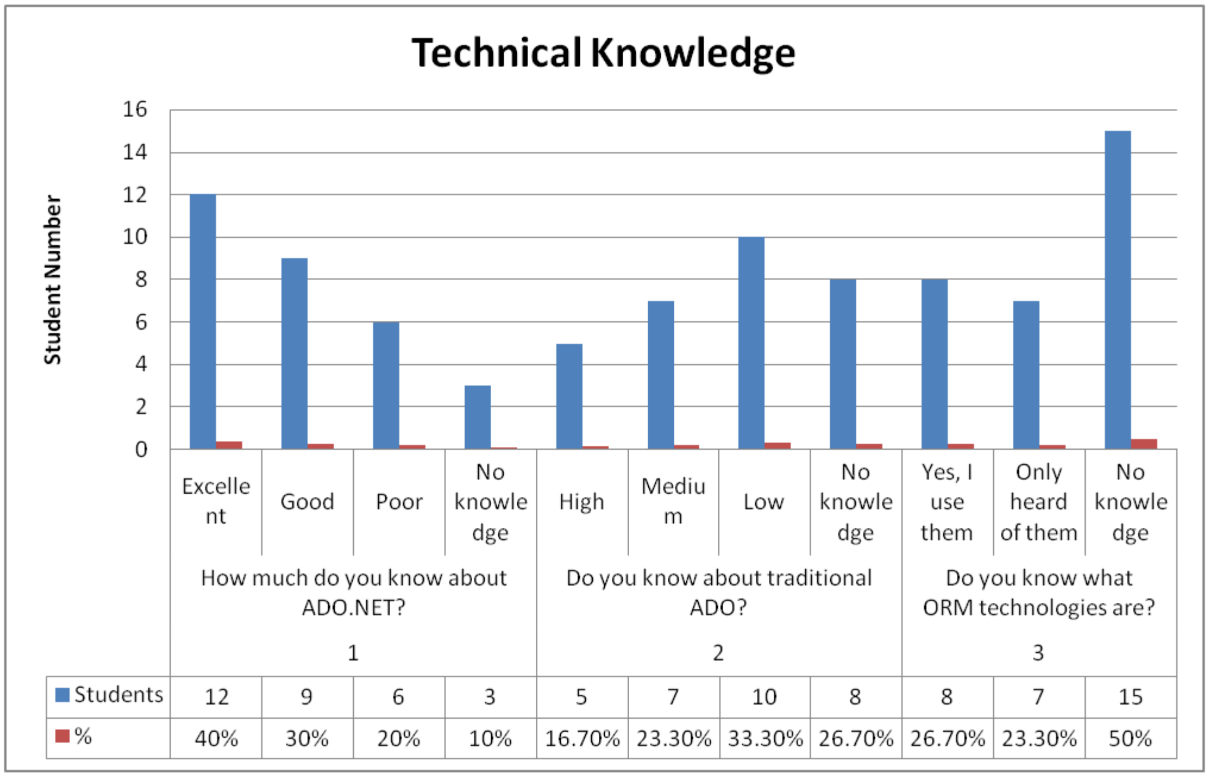
The descriptive survey approach was selected for this study due to its suitability for the objectives and nature of the research.A sample of thirty students from the Higher Institute of Science and Technology Alshmokh, Tripoli's Computer Technologies Department participated in a field study. Students working on graduation projects made up the sample, which was chosen at random to assess how well ADO.NET was used in comparison to other systems like conventional ADO and ORM.

8.2 Sample Questionnaire:

The questionnaire was distributed to 30 students from the Computer Technologies Department at the Higher Institute of Science and Technology Alshmokh, Tripoli. The survey included 15 questions divided into 5 main categories: Technical Knowledge, Learning Ease, Performance Evaluation, Practical Preferences, and Flexibilityas shown below.

Table 1: Technical Knowledge

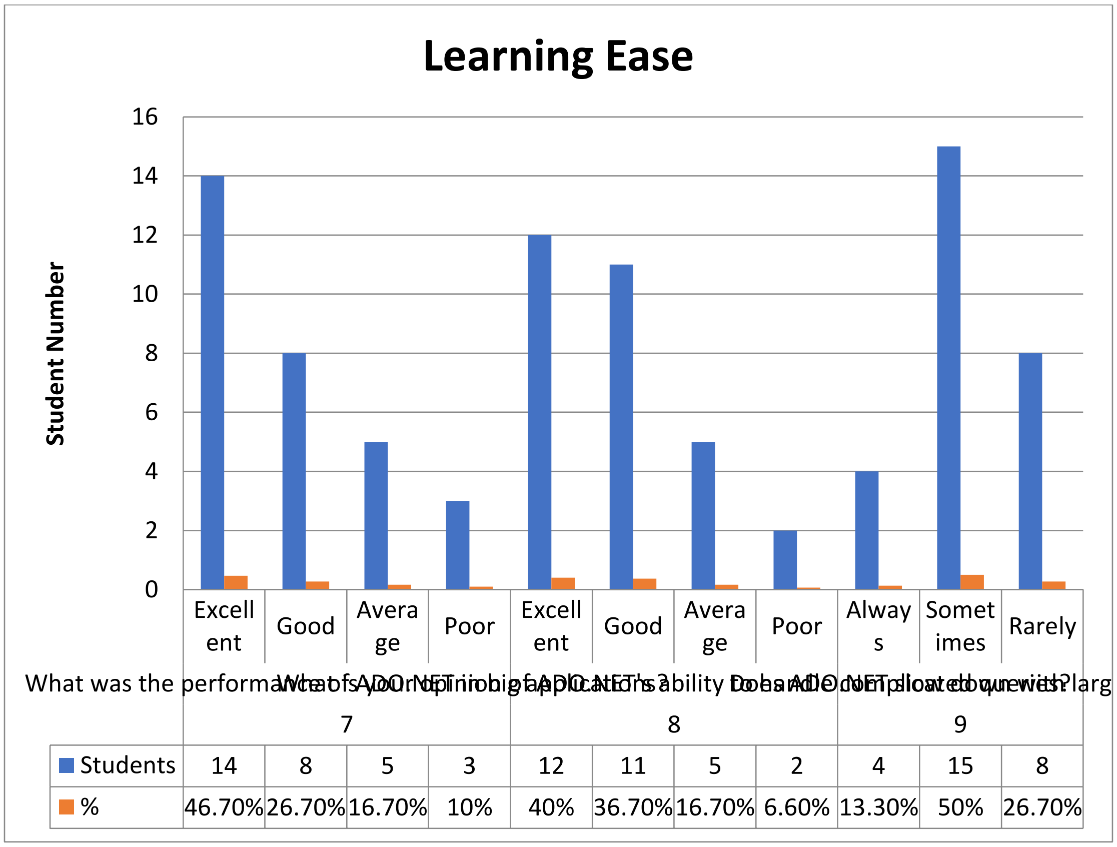
Q#	Question	Options	Students	%
1	How much do you know about ADO.NET?	Excellent	12	40%
		Good	9	30%
		Poor	6	20%
		No knowledge	3	10%
2	Do you know about traditional ADO?	High	5	16.7%
		Medium	7	23.3%
		Low	10	33.3%
		No knowledge	8	26.7%
3	Do you know what ORM technologies are?	Yes, I use them	8	26.7%
		Only heard of them	7	23.3%
		No knowledge	15	50%



	Excellent	Good	Poor	No knowledge	High	Medium	Low	No knowledge	Yes, I use them	Only heard of them	No knowledge
🟦 Students	12	9	6	3	5	7	10	8	8	7	15
🟥 %	40%	30%	20%	10%	16.70%	23.30%	33.30%	26.70%	26.70%	23.30%	50%

Table 2: Learning Ease

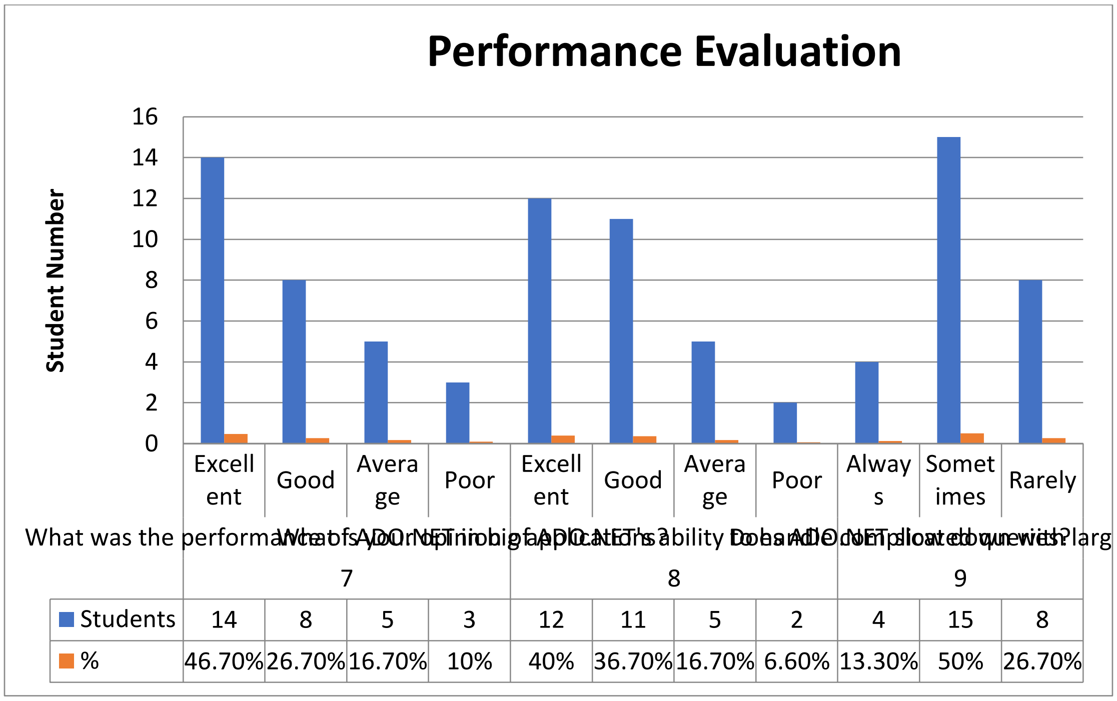
Q#	Question	Options	Students	%
4	How difficult is it to learn ADO.NET compared to ADO?	Much easier	18	60%
		Somewhat easier	6	20%
		Similar	4	13.3%
		More difficult	2	6.7%
5	What do you like most about ADO.NET?	Support for different models	22	73.3%
		Good documentation	15	50%
		Practical examples	12	40%
6	What was the hardest part of learning ADO.NET?	Database connection complexity	10	33.3%
		Lack of practical examples	8	26.7%
		Technical terminology difficulty	7	23.3%
		No difficulties	5	16.7%



	7				8				9		
	Excellent	Good	Average	Poor	Excellent	Good	Average	Poor	Always	Sometimes	Rarely
■ Students	14	8	5	3	12	11	5	2	4	15	8
■ %	46.70%	26.70%	16.70%	10%	40%	36.70%	16.70%	6.60%	13.30%	50%	26.70%

Table 3: Performance Evaluation

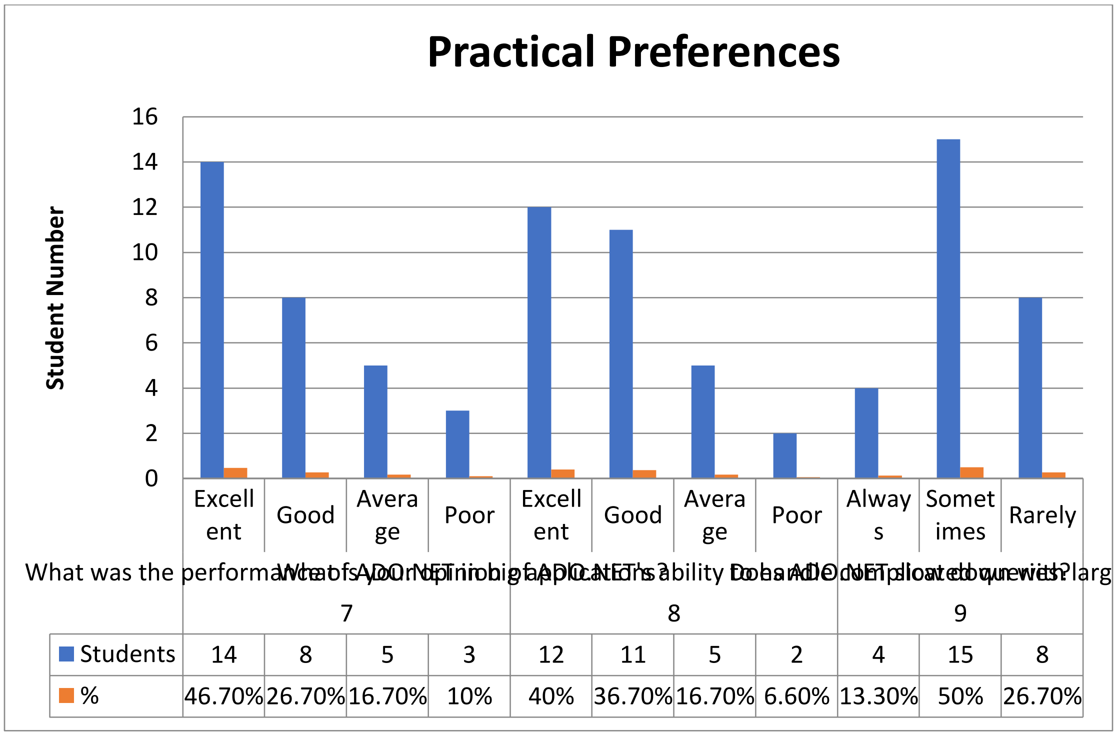
Q#	Question	Options	Students	%
7	What was the performance of ADO.NET in big applications?	Excellent	14	46.7%
		Good	8	26.7%
		Average	5	16.7%
		Poor	3	10%
8	What is your opinion of ADO.NET's ability to handle complicated queries?	Excellent	12	40%
		Good	11	36.7%
		Average	5	16.7%
		Poor	2	6.6%
9	Does ADO.NET slow down with large data?	Always	4	13.3%
		Sometimes	15	50%
		Rarely	8	26.7%
		Never	3	10%



	Excellent	Good	Average	Poor	Excellent	Good	Average	Poor	Always	Sometimes	Rarely
🟦 Students	14	8	5	3	12	11	5	2	4	15	8
🟧 %	46.70%	26.70%	16.70%	10%	40%	36.70%	16.70%	6.60%	13.30%	50%	26.70%

Table 4: Practical Preferences

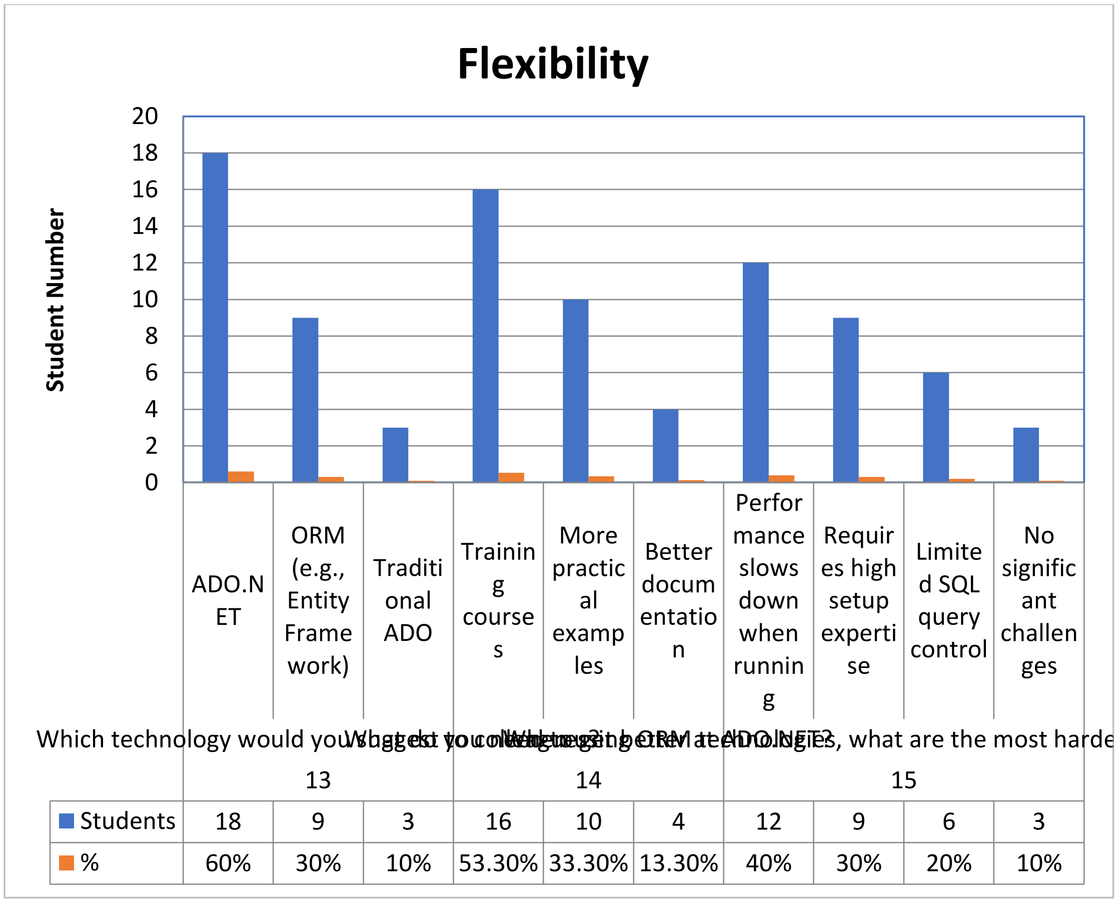
Q#	Question	Options	Students	%
10	For projects, which technology do you prefer?	ADO.NET	19	65%
		ADO	5	16.7%
		ORM	6	20%
11	Which feature of ADO.NET do you think is better than traditional ADO?	Execution speed	14	46.7%
		Maintenance ease	9	30%
		Modern system support	7	23.3%
12	How many projects have you actually used ADO.NET in?	More than 3 projects	6	20%
		2-3 projects	12	40%
		One project	8	26.7%
		Never used	4	13.3%



	7				8				9		
	Excellent	Good	Average	Poor	Excellent	Good	Average	Poor	Always	Sometimes	Rarely
🟦 Students	14	8	5	3	12	11	5	2	4	15	8
🟧 %	46.70%	26.70%	16.70%	10%	40%	36.70%	16.70%	6.60%	13.30%	50%	26.70%

Table 5: Flexibility

Q#	Question	Options	Students	%
13	Which technology would you suggest to colleagues?	ADO.NET	18	60%
		ORM (e.g., Entity Framework)	9	30%
		Traditional ADO	3	10%
14	What do you need to get better at ADO.NET?	Training courses	16	53.3%
		More practical examples	10	33.3%
		Better documentation	4	13.3%
15	When using ORM technologies, what are the most hardest things for you?	Performance slows down when running	12	40%
		Requires high setup expertise	9	30%
		Limited SQL query control	6	20%
		No significant challenges	3	10%



	13			14			15			
	ADO.NET	ORM (e.g., Entity Framework)	Traditional ADO	Training courses	More practical examples	Better documentation	Performance slows down when running	Requires high setup expertise	Limited SQL query control	No significant challenges
🟦 Students	18	9	3	16	10	4	12	9	6	3
🟧 %	60%	30%	10%	53.30%	33.30%	13.30%	40%	30%	20%	10%

8.3 Sample Analysis

The following important conclusions were drawn from the field study that was carried out on a sample of thirty students.

1. Technical Knowledge

40% of students have excellent knowledge of ADO.NET, while 10% have no knowledge at all.

Regarding traditional ADO, 33.3% have low knowledge, and 26.7% have no knowledge.

Half of the students (50%) are not familiar with ORM technologies, while 26.7% actively use them.

2. Learning Ease

60% find learning ADO.NET much easier compared to traditional ADO.

The most liked features of ADO.NET are support for different models (73.3%) and good documentation (50%).

The biggest challenges in learning were database connection complexity (33.3%) and lack of practical examples (26.7%).

3. Performance Evaluation

46.7% rated ADO.NET performance in large applications as excellent.

40% consider ADO.NET's ability to handle complex queries as excellent, while 50% experience slowdowns sometimes with large data sets.

4. Practical Preferences

65% prefer using ADO.NET in their projects, compared to 16.7% for ADO and 20% for ORM.

The top advantage of ADO.NET over traditional ADO is execution speed (46.7%).

60% of students have used ADO.NET in two or more projects.

General Conclusion:

The results show that ADO.NET is well known and easier to learn compared to other technologies. Its performance in large projects is satisfactory for most students. However, students need more support to understand some technical concepts and more practical examples to facilitate learning.

8.4 Comparison of ADO.NET with Other Systems

The table5 presents a comparison between ADO.NET, traditional ADO, and ORM (Object-Relational Mapping) across several important criteria for database management:

Evaluation Criteria	ADO.NET	Traditional ADO	ORM
Technical Knowledge	40% excellent 10% no knowledge	33.3% low knowledge 26.7% no knowledge	50% not familiar 26.7% active users
Learning Ease	Rated as easier than ADO by 60% of students. Main advantages: multi-model support (73.3%) and good documentation (50%). Main challenges: DB connection complexity (33.3%).	Simpler conceptually but limited. Relies on direct, persistent connections, lacking support for disconnected or XML features.	Harder to learn initially. 50% of students not familiar with ORM concepts. Provides abstraction through object-relational

Evaluation Criteria	ADO.NET	Traditional ADO	ORM
			mapping.
Performance in Large Apps	46.7% rated performance as excellent in large apps. Strong in query handling but slower with very big data (50%).	Poor efficiency in large or complex apps. Performance issues due to constant connection reliance.	Efficient in coding simplicity and reducing errors, but sometimes less performant with complex/big queries
Flexibility	Balanced solution, used in 60% of projects, supports connected/disconnected models and XML integration.	Very limited flexibility, only connected model. Rarely preferred in modern projects (16.7%).	Highly flexible, supports multiple databases and reduces dependency on SQL. Preferred in 20% of projects, better for long-term use.

Table 5: Comparative Analysis of ADO.NET, Traditional ADO, and ORM Technologies

9.Conclusions and Recommendations:

The study shows that ADO.NET is a practical and effective solution for the Libyan labor **market**, especially for organizations that struggle with unstable internet connections, managing large volumes of data, or developing practical skills among local developers. Its hybrid architecture allows data to be processed locally before synchronization, reducing reliance on continuous internet access. This makes it easier for institutions handling accounting, inventory, or customer follow-up to adopt approaches like the Disconnected Model, Batch Updates, and Retry Logic without major disruptions.

ADO.NET also performs efficiently with large and complex applications while minimizing network load. Governmental and financial institutions, for example, can benefit from techniques like Paging to divide data, optimizing SQL queries, and caching frequently used data. While it is generally easier to learn and document than alternatives, practical, hands-on examples are still essential. Integrating ADO.NET into academic programs, offering practical workshops, and providing clear, scenario-based documentation can help bridge this gap.

Additionally, ADO.NET provides greater control and better performance than ORM tools, making it well-suited for large-scale projects that demand precision and stability, such as banking, government, or industrial applications. In essence, ADO.NET is more than just a technical tool—it is a comprehensive solution that helps Libyan organizations improve performance, manage large databases effectively, and build the skills of their workforce.

Based on the findings of this study, it is recommended that organizations adopt ADO.NET in a structured and systematic manner, emphasizing practical, real-world applications. Continuous training should be provided for developers, and the technology should be integrated into higher education curricula to build local expertise. Best practices should be implemented to manage connectivity challenges and ensure optimal performance. Projects that demand high efficiency, reliable data management, and full control should prioritize the use of ADO.NET. Following these recommendations will help organizations improve operational efficiency, enhance service quality, and prepare effectively for future technical challenges.

10.Reference:

1. Sivakumar, S., & Arulprakash, P. (2018). *Comparative Analysis of ADO and ADO.NET*. International Journal of Advanced Research in Computer Science, 9(2), 58-62.
2. Tiwari, S. (2021). *Performance Trade-offs in ORM vs. Raw SQL*. Journal of Systems Architecture, 118, 102210.
3. Ahmed, T., & Majeed, K. (2020). *ADO.NET in Educational Systems: Efficiency and Security*. Journal of Educational Technology Systems, 49(1), 98-115.
4. Esposito, D. (2019). *Programming Microsoft .NET Core*. Microsoft Press. (Ch. 6: "Data Access with ADO.NET")
5. Richter, J. (2020). *CLR via C#* (5th ed.). Microsoft Press. (Sec. "ADO.NET Architecture")
6. Garcia-Molina, H. (2019). *Database Systems: The Complete Book* (3rd ed.). Pearson. (Ch. 10: "Disconnected Data Access")
7. Dreheeb, A. M., & El Tajouri, H. (2025). Role Of Artificial Intelligence In Enhancing Cyber Security. Bani Waleed University Journal of Humanities and Applied Sciences, 10(3), 121-129.
8. Creswell, J. W., & Creswell, J. D. (2018). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (5th ed.). SAGE. (Ch. 8: "Survey Designs")
9. Microsoft. (n.d.). *ADO.NET Overview*. Microsoft Learn. Retrieved July 28, 2025, from https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/
10. TutorialsPoint. (n.d.). *ADO.NET Tutorial*. Retrieved July 28, 2025, from https://www.tutorialspoint.com/ado.net/index.htm
11. Almarimi, A. F., & Salem, A. M. (2025). Machine Learning using Simple Linear Regression. Bani Waleed University Journal of Humanities and Applied Sciences, 10(3), 178-184.
12. Zhou Dongxiang, Wu Jinlu. (2025). : ADO.NET + *SqlDataSource* + *LINQ*. Books.com.tw. https://www.books.com.tw/products/0010887715
13. Microsoft. (2025). ADO.NET *Overview*. Microsoft Docs. https://learn.microsoft.com/zh-tw/dotnet/framework/data/adonet/ado-net-overview
14. Microsoft. (2025). *LINQ to SQL Reference*. Microsoft Docs. https://learn.microsoft.com/zh-tw/dotnet/framework/data/adonet/sql/linq/reference
15. Microsoft. (2025). ADO.NET *Architecture*. Microsoft Docs. https://learn.microsoft.com/zh-cn/dotnet/framework/data/adonet/
16. Eric Zhou. (2025). *SQL Parameterization in* ADO.NET. Eric Zhou. https://www.cnblogs.com/tianqing