

# Comparative Performance Analysis

## ADO.NET, Entity Framework, and MongoDB

This study evaluates the performance of three distinct database frameworks — ADO.NET, Entity Framework, and MongoDB — through CRUD (Create, Read, Update, Delete) operations across various dataset sizes.

## Summary of Findings

**MongoDB** is the most efficient framework, showcasing consistent performance across all CRUD operations, particularly with smaller datasets. However, deletion times for larger datasets require optimization to maintain flexibility.

**ADO.NET** demonstrates superior performance with larger datasets. It is well-suited for applications handling large volumes of data.

**Entity Framework** exhibits longer operation times compared to the other frameworks, regardless of dataset size. This highlights the need for careful performance evaluation when selecting a database framework for projects.

## Testing Approach

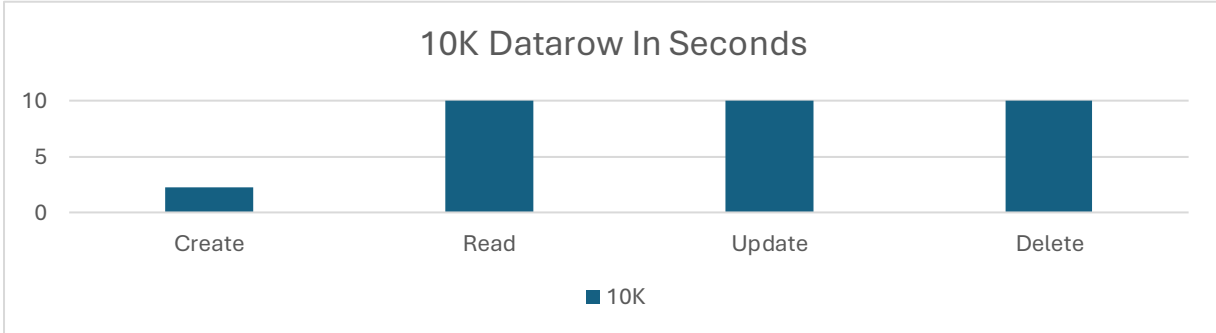
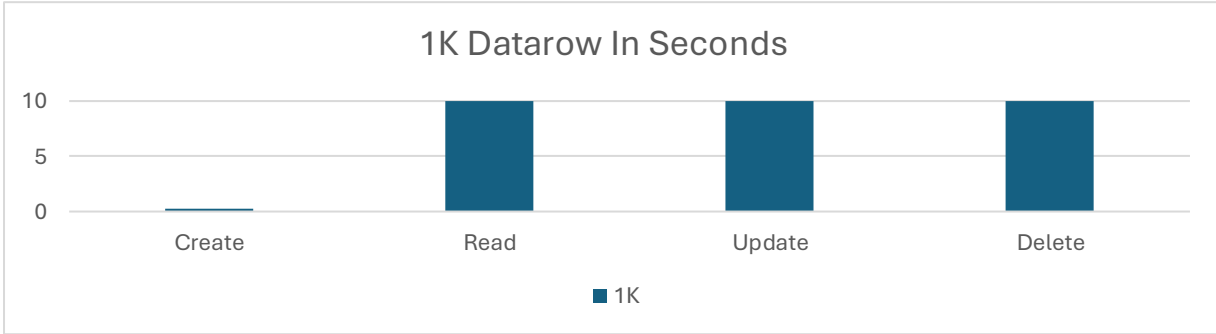
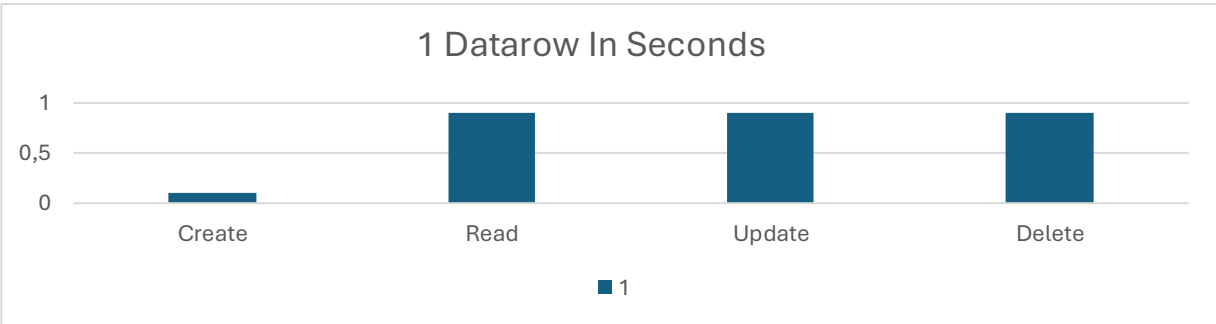
This performance analysis involved multiple testing rounds to ensure a comprehensive evaluation of each framework's capabilities and limitations. This approach provided a deep understanding of the frameworks' performance characteristics across various scenarios and dataset sizes. **Importantly, all databases contained existing data during testing**, ensuring that the queries were tested in realistic conditions rather than against empty databases.

# Detailed Analysis

## ADO.NET Performance

The enchantedears database, containing 8,407,375 records, was used for testing CRUD operations with an artist named Joe Jonas. A for loop added numbers to the artist's name to ensure unique entries. Results showed efficient handling of up to 1 million records, particularly in data creation. To run the queries I have used the SQL Manager application, but also checked out using Visual Studio 2022 for checking the accuracy of the data.

	Create	Read	Update	Delete
1	0.1 seconds	0.9 seconds	0.9 seconds	0.9 seconds
1 000	0.3 seconds	10 seconds	10 seconds	10 seconds
10 000	2.2 seconds	10 seconds	10 seconds	10 seconds
1 000 000	218.8 seconds	21 seconds	18 seconds	17 seconds

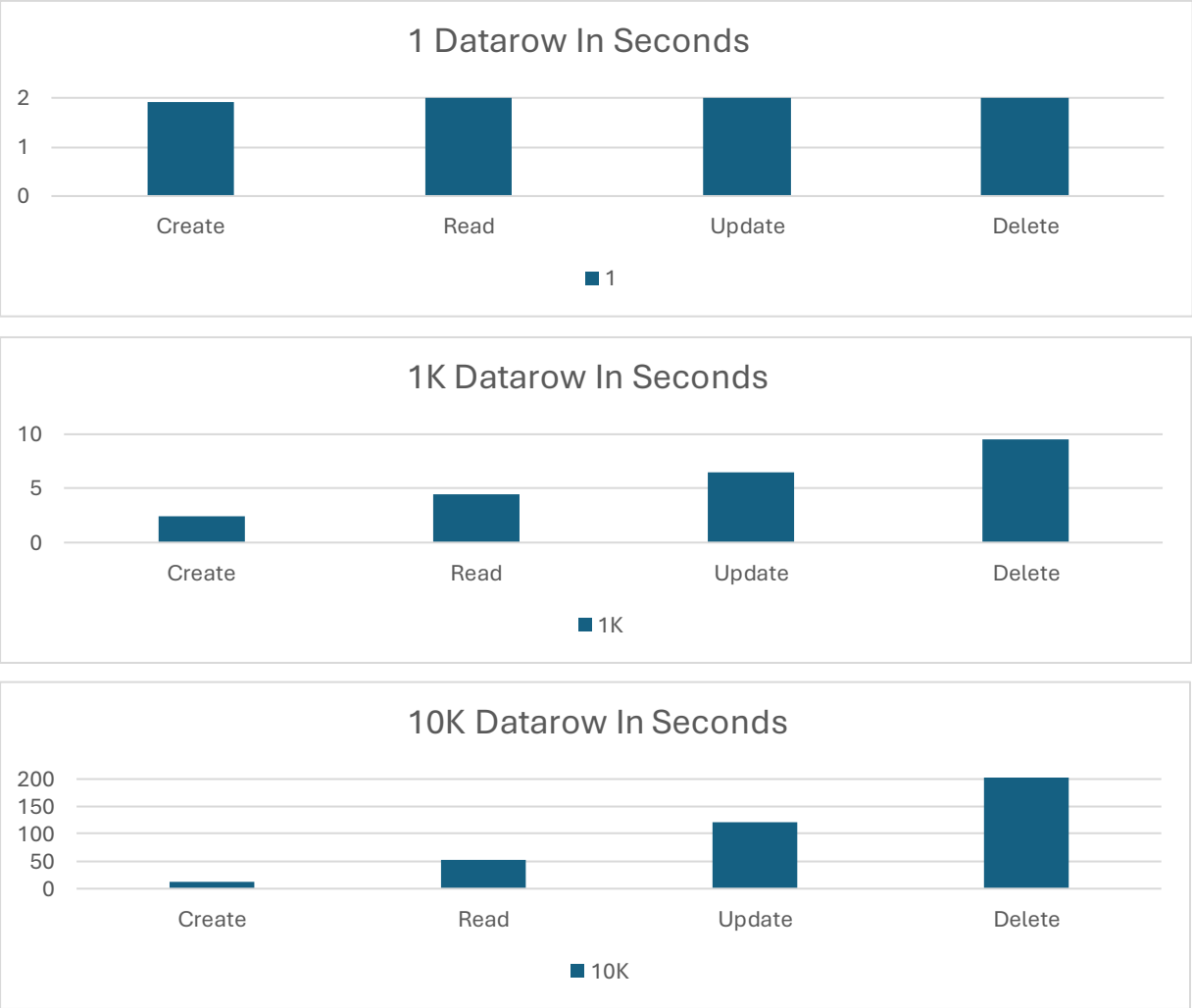


## Entity Framework Performance

Entity Framework testing involved adding Taylor Swift as a pop artist to a database containing 379,966 records. Performance issues were noted as dataset size increased, especially during insertion operations. To run the queries I have used the Visual Studio 2022.

	Create	Read	Update	Delete
1	1.9 seconds	2.0 seconds	2.0 seconds	2.0 seconds
1 000	2.4 seconds	4.5 seconds	6.5 seconds	9.5 seconds
10 000	13.4 seconds	53.0 seconds	122.4 seconds	224.6 seconds
1 000 000	1 029.4 s	2 135.6 s	? seconds	? seconds

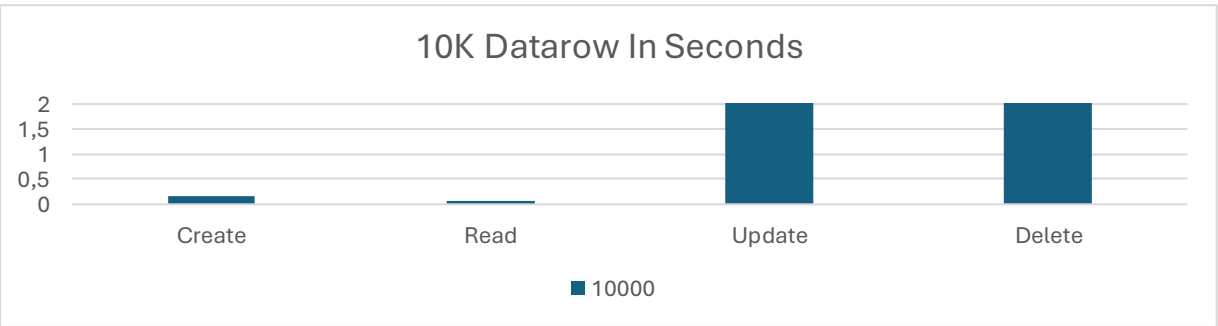
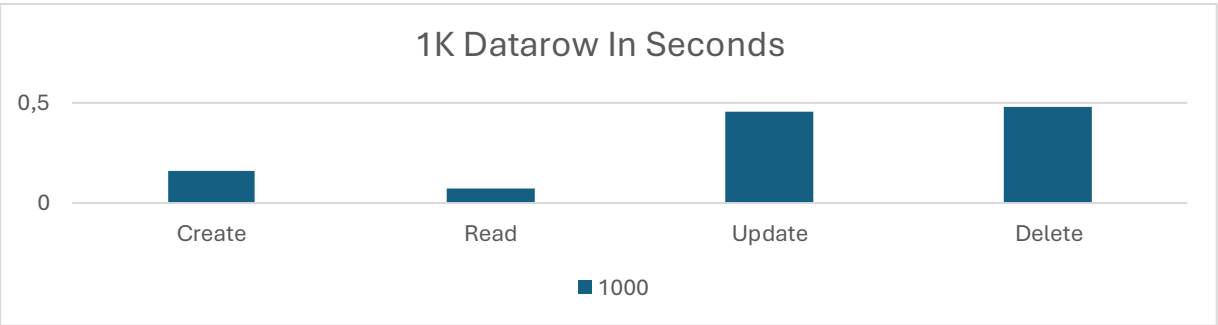
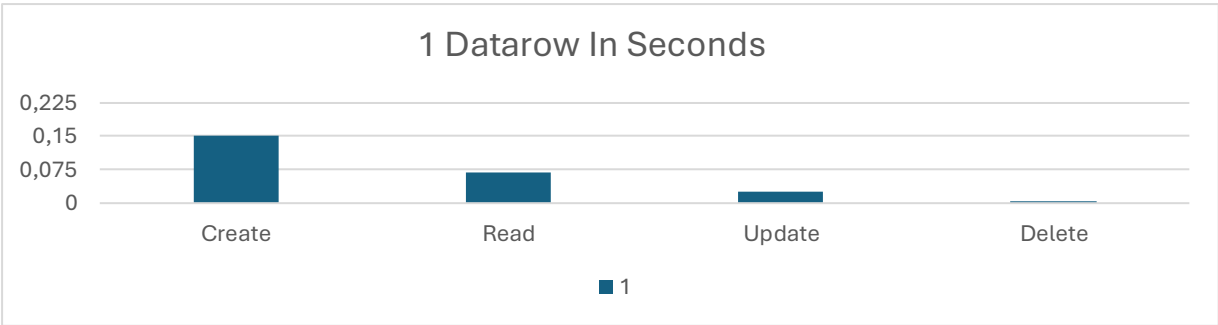
*It took a while but after 8 hours it did not finish, and even in the morning it did not get updated or removed after 7 hours. It can be seen it runs but it takes a lot of times, especially with updating and removing 1 million data within a database that contains less than 500K data.*



## MongoDB Performance

Testing MongoDB involved adding "The Tortured Poets Department" album by Taylor Swift. MongoDB demonstrated fast update and delete operations, though insertion and retrieval times increased with larger datasets. The database contained 2,090,121 records during testing. To run the queries I have used MongoDB application, but also tested out within the Visual Studio 2022 for checking how accurate the data is.

	Create	Read	Update	Delete
1	0.152 seconds	0.07 seconds	0.025 seconds	0.005 seconds
1 000	0.163 seconds	0.076 seconds	0.458 seconds	0.476 seconds
10 000	0.156 seconds	0.071 seconds	3.296 seconds	4.12 seconds
1M	0.155 seconds	0.111 seconds	338.664 seconds	989.947 seconds



## Conclusion

This comprehensive analysis underscores the importance of selecting the right database framework based on specific application requirements and dataset sizes.

**MongoDB** is highly efficient for a broad range of CRUD operations, especially with smaller datasets, but requires optimization for large-scale deletions.

**ADO.NET** offers robust performance for large datasets, making it a suitable choice for applications with high data volume requirements.

**Entity Framework** shows slower performance, emphasizing the need for careful consideration of its use in projects where performance is critical.

