

Comparison of C# and Python programming languages in terms of performance and coding on SQL server DML operations

ABSTRACT

Nowadays, there are several computer programming languages and database management systems, and they have advantages and disadvantages one over another. Databases are essential components of computer programs, regardless of their language. Thanks to databases, computer programs record the data that they need or produce and perform the given tasks by retrieving these data when necessary. For a computer program to run efficiently and quickly, it is essential that both the database and the computer program are well structured. In this study, C# and Python languages, which are among the most widely used programming languages out of these various programming languages, have been evaluated in terms of transaction performance and the amount of code that needs to be written to perform SQL DML (Data Manipulation Language) operations such as INSERT, SELECT, UPDATE, DELETE operations and print the results of the operation on the screen via MSSQL database,¹ which is one of the most widely used database management systems. In terms of processing performance, it was observed that neither programming language provided a significant superiority over the other, although mathematically Python language seems to have performed better when looking at the processing times. In terms of code size and readability, although C# is generally considered to be a more readable language in terms of code readability, in the context of the programs written for this study, there was no difference between the two programming languages in terms of code readability. In terms of code size, Python provided a clear superiority. As a result, it has been determined that both languages have superior features compared to each other, and that there is no definite superiority between these two languages that can be a reason for preference over each other in DML operations. The choice of language should be based on the requirements of the project, the ecosystem and the skills of the team.

Keywords: C#, Python, MSSQL, DML Operations, Code Writing, Performance

INTRODUCTION

Today, information and communication technologies such as computers, mobile devices, the internet, and many other digital communication tools are widely used, and these technologies deeply affect social, cultural, economic, and social life. One of the first concepts that comes to mind when it comes to information and communication technologies is data, and the other is computer software. For this reason, storing data in a database system through computer software, and retrieving, correcting, or deleting data when necessary constitutes the basis of information and communication technologies. In modern software development processes, performing database operations effectively and efficiently is a critical factor affecting the performance and reliability of applications. Therefore, the choice of programming language has a direct impact on the effectiveness of the tools used to perform database operations. C# and Python are currently among the most widely used programming languages, and both are used to interact with relational database management systems such as SQL Server. This study aims to make a comparison between C# and Python programming languages in terms of performance and code writing efficiency in SQL Server Data Manipulation Language (DML) operations.

One of the main goals of this study is to compare the performance of C# and Python in SQL Server DML transactions, to determine the impact of each language in terms of transaction execution time, and to evaluate the impact of each language in terms of performance and code quality, such as code readability, flexibility, and error handling capabilities.

Based on the results obtained, it will be possible to identify the advantages and disadvantages of C# and Python in the creation and execution of queries for SQL Server DML transactions and analyze whether each language is more suitable in specific use cases.

DATABASE MANAGEMENT SYSTEMS

Today, human life is intertwined with information technologies and there is almost no area where information devices are not available. From daily activities to working life, from health to entertainment, computers, software and therefore databases and computer programs are integrated with human beings. This integration has led to many differences in the quantity and quality of data to be stored and used, and in the way data is retrieved. This has led to different approaches and systems

**Mahmut
YEŞİLYURT**¹



¹ Department of Computer Technologies, İzmir Kavram Vocational School, İzmir, Türkiye

Yusuf Ziya AYIK^{2,3}



² Department of Management Information Systems, Graduate School of Social Sciences, Atatürk University, Erzurum, Türkiye

³ Department of Computer Technologies, Vocational School of Technical Sciences, Atatürk University, Erzurum, Türkiye



Received 03.06.2024
Accepted 24.06.2024
Publication Date 30.06.2024

Corresponding author:

Mahmut YEŞİLYURT

E-mail:

umyesilyurt@gmail.com

Cite this article: Yeşilyurt M, Ayık Y S.

Comparison of C# and Python programming languages in terms of performance and coding on SQL server DML operations. *NanoEra*.

2024;4(1):23-33



Content of this journal is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

in database systems and data models.

Database Management System (DBMS) is a collection of data categorized in a table or several tables integrated with each other and an application program that regulates how to access this data.² This collection of data is often called a database. The main purpose of DBMS is to provide a way to quickly store and retrieve the information in the database. Databases allow stored data to be queried in relation to other data that is related to that data. In a relational database, data is stored in the form of relationships or two-dimensional tables, and there is a data relationship between the tables.³

While the amount of data to be stored is increasing day by day, the structure of the data to be stored is of great importance in database design. In addition to the structured data stored in a certain template, the widespread use of the internet today has disrupted the traditional data structure and led to the emergence of unstructured databases. In this context, it would be more accurate to examine the database model in two groups: relational databases, where structural data is stored within a template, and non-relational databases, where data that is not within a certain template is stored, or, in other words, NoSQL databases, which do not have to be relational.

Relational Database Systems

Relational Database Systems (RDMS) Organization based on the relational data model, first proposed by Edgar Frank Codd in 1970 are various software systems used to manage database systems.⁴ Although traditional database systems such as the network data model and the hierarchical data model were used before relational databases, their use has declined with the development and more widespread use of relational database systems.

Relational database systems are the most preferred database management system for storing structured data whose structure is defined on a specific template.⁴ In relational database systems, data is stored in tables consisting of rows and columns. Each transaction in relational databases is defined as a transaction, and these transaction operations work in an all-or-nothing context. Each transaction must have basic properties known as ACID. Transactions that do not have ACID basic properties are not executed and are rolled back. ACID stands for Atomic, Consistent, Isolated, and Durable.⁵

ACID principles;⁵

- Atomic: guarantees the successful completion of an operation in the database, such as Insert, Update or Delete. The initial operation cannot be split; it must either be committed, i.e., applied, or rollback, i.e., canceled.
- Consistent: it guarantees that the data stored in the database conforms to the predetermined template, that is, its consistency. When a transaction is committed or rolled back, the database must maintain its consistency. If the transaction completes successfully, the changes are applied to the database, if it fails, it is automatically rolled back.
- Isolation: each transaction operation is performed in isolation, meaning that no other transaction operation can

access the result of this transaction operation until the transaction is finalized.

- Durable: when a transaction is successfully completed, i.e., committed, the result of the transaction is persistent in the database, no matter what.

The most popular relational databases worldwide are Oracle, MySQL, Microsoft SQL Server and PostgreSQL.⁶

NoSQL Database Systems

NoSQL database Systems process large volumes of rapidly changing data in unstructured form in ways that differ from a relational database containing rows and tables. NoSQL technologies have been used under various names since the 1960s. Due to the changes in the data environment, the popularity of NoSQL database systems is increasing day by day as software developers have to work with large volumes and a wide variety of data generated by cloud technologies, social media platforms, and mobile devices and have to adapt to change.⁷

Data from different types of data-sending sources, such as social media applications, third-party databases, mobile devices, and smart sensors, is not suitable for storage in a relational model database system, and this is where non-relational NoSQL database systems come into play.⁸

The term "Not only SQL" is a phrase often used for NoSQL databases. This term emphasizes an approach that is not limited to SQL queries only, as in traditional relational database systems. In this way, it aims to take full advantage of the flexibility that NoSQL databases offer.

NoSQL database systems operate on the basis of certain principles to meet the requirements of big data and distributed systems, called BASE which represent the following concepts.⁹

- Basically Available: This principle emphasizes the flexibility of the system. NoSQL databases usually offer extensibility at a high scale and provide extensibility with the flexibility to work in the system. Certain data records may sometimes be temporarily unavailable, but generally, the system will be functional.
- Soft state: This principle refers to consistency and flexibility. NoSQL databases offer a more flexible approach to consistency and do not require immediate synchronization between copies of data. This is critical for expansion on a larger scale.
- Eventually Consistent: This policy complements the consistency and flexibility of NoSQL databases. It states that consistency will eventually be achieved after changes, but this process may not happen instantly. In this case, there may be temporary inconsistency between copies of data, but eventually the system will reach the desired state of consistency.

These features demonstrate that NoSQL databases offer a different approach than traditional relational databases and are better suited to modern applications that require scalability and flexibility.

NoSQL Database Types

- Document-based NoSQL Databases: These types of databases are used to store and query data in the form of documents. For example, MongoDB stores documents in JSON or BSON format. These types of databases usually allow flexible data structures and dynamic schemas.
- Column-based NoSQL Databases: Column-based databases are ideal for large amounts of horizontally scaled data. Examples such as HBase and Cassandra are column-based databases. They are designed to provide high performance and scalability.
- Key-value based NoSQL Databases: These types of databases store data in the form of simple key-value pairs and are ideal for applications that require fast access. Databases such as Redis and Amazon DynamoDB are key-value-based NoSQL solutions.
- Graph-based NoSQL Databases: These databases are used to store and query relational data. Neo4j is an example of this type of database. Graph-based databases are ideal for effectively managing complex relationships and connections.
- Each type of NoSQL offers different advantages for specific use cases and requirements. It is important to choose the most suitable one according to the needs of the application and the data structure.
- The most popular relational databases worldwide are MongoDB, Redis, Cassandra, Neo4j .^{1,10}

What Are The Most Used DMSs?

Stack Overflow, a question-and-answer website about computer programming with more than 50 million users worldwide,

published the following graph about the most commonly used DMSs in the most popular technologies section of the "Developer Survey" conducted in 2022.

As seen in this survey of 63,327 computer programmers (Fig 1), in which participants can choose more than one database system, the database systems with the highest preference rates are MySQL, PostgreSQL, SQLite, MongoDB and Microsoft SQL Server.

Python and C# Programming Languages

Python Programming Language

Python Overview

Python is a general-purpose, interactive, and high-level programming language. It was started to be developed by Guido van Rossum in 1991. Its basic philosophy is code readability and simplicity.¹¹

Python is a programming language that has been accepted worldwide and has had a wide user base since its first release. It is used both in academia and industry, from scientific research to financial analysis.

Technology giants frequently use Python in areas such as product development and data analysis. Python's popularity is also supported by the fact that it has become part of university courses. Its use as a core language in programming courses encourages students to learn Python and familiarize themselves with its use in their future professional lives.

Python's powerful standard library and wide module support make it usable in a variety of fields. Thanks to these features, it can be used effectively in data analysis, web development, machine learning, and artificial intelligence, and many more. Python's flexibility and easy readability make it preferable for both experienced developers and beginners.¹²

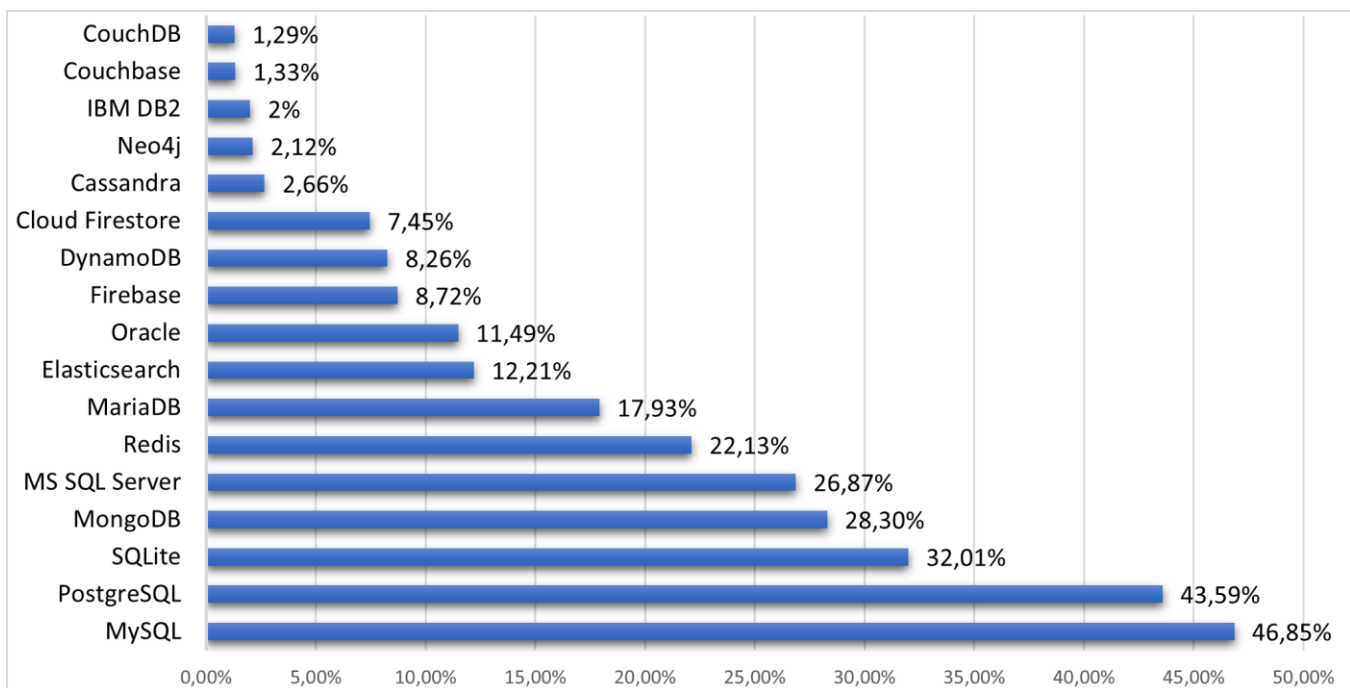


Fig 1. Survey of the most popular database systems (May-22)¹

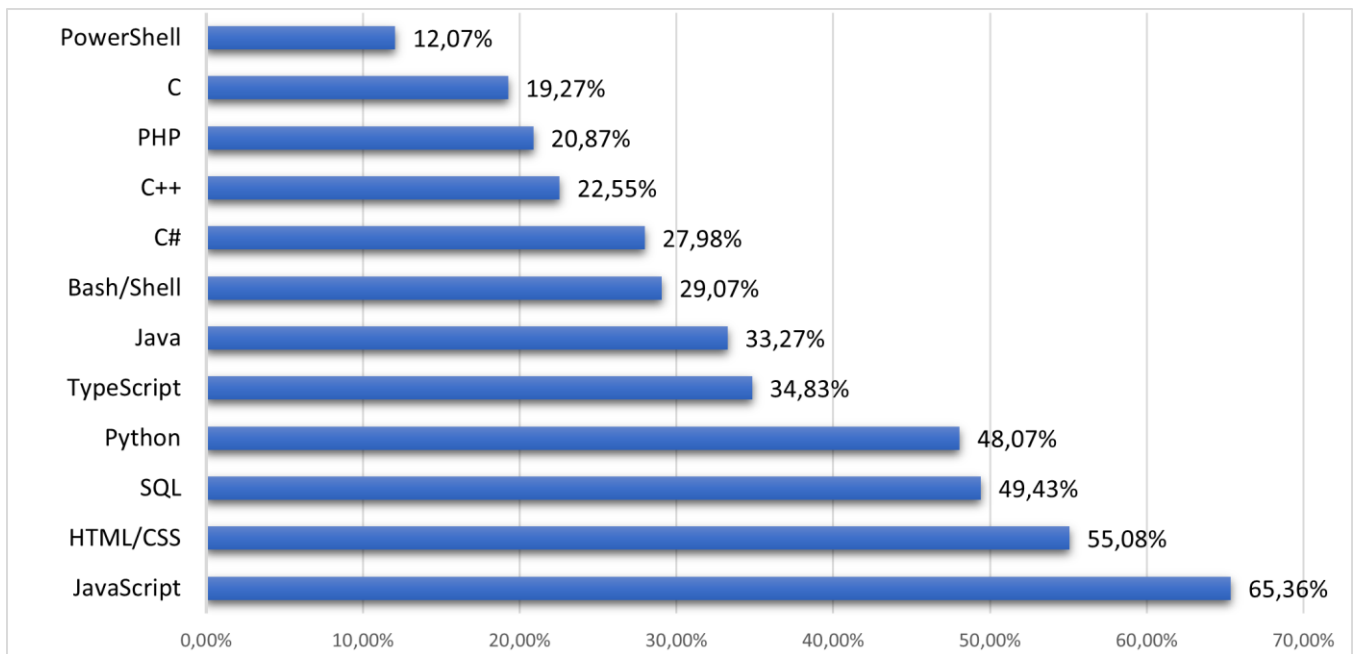


Fig 2. Survey of the most popular programming languages (May-22)¹

Figure 2. as can be seen in 'Python', Python has become one of the most popular programming languages.

We can attribute this to the following features of the Python programming language.

- Simple and Readable Syntax: Python's simple and organized syntax makes the code easy to understand.
- High Level Python is a high-level programming language, meaning that it is user- friendly and does not require dealing with complex details.¹¹
- Various Modules and Libraries: Python has a very rich standard library and a large ecosystem of third-party libraries that can be used in various domains.¹³
- Portability: Python can run on many platforms (Windows, Linux, macOS) and offers flexibility in portability.¹⁴
- Object Oriented: Python supports the object-oriented programming paradigm, meaning it can use classes and objects to model real-world entities.¹⁵
- Broad Community Support: Python has a large and vibrant community of users worldwide, enabling questions to be answered quickly and solutions to be found.¹⁵

Areas of Use

Python's use cases are quite broad and include.¹²

- Web Development (Django, Flask)
- Data Science and Artificial Intelligence (NumPy, Pandas, TensorFlow, PyTorch)
- Machine Learning (Scikit-learn)
- Computer Vision (OpenCV)
- Game Development (Pygame)
- REST API and Web Framework Creation
- Python

- Network Programming
- Automation and Scripting
- Scientific Calculations
- Financial Analysis
- Web Reser (Beautiful Soup, Scrapy)

Advantages and Disadvantages Compared to Other Programming Languages

Advantages of Python:

- Readable and simple syntax
- Extensive library and module support
- Usability on various platforms
- Ideal for rapid prototype development
- It has a large and active community

Disadvantages of Python:

- Slow execution speed: May run slower than languages like C or C++
- Not suitable for mobile application development
- There is no strong type system, so error detection can be difficult in large- scale projects

C# PROGRAMMING LANGUAGE

C# Overview

C# is a powerful and modern programming language developed by Microsoft with an open-source development environment and is typically used for the Windows platform. C# was developed by Microsoft in the late 1990s, inspired by Java, C++ and other programming languages, and became the core language of the .NET platform. Developed by Anders Hejlsberg and his team, who also partially founded the Delphi language, the language was officially released as C# 1.0 in 2002.¹⁶ C# 2.0 was released with Visual Studio 2005, especially with the

language's extended language features, generics, nullable types, anonymous methods, and other enhancements. C# 3.0 was released with the .NET Framework 3.5 with the Language Integrated Query (LINQ) feature, which makes data manipulation easier and more natural. C# 5.0 in 2012 and C# 7.2 in 2017 included important language improvements such as `async/await`, pattern matching, nullable reference types, and local functions. Since 2019, this language has continued to evolve and is now used as C# 13, which was released with the current version .NET 9.¹⁷

Here are some of the features that make the C# language powerful;¹⁷

- **Object Oriented Programming (OOP):** C# is based on the object-oriented programming paradigm. This refers to an approach where data and functions are organized into classes, and interaction between these classes is enabled. OOP principles increase code reusability, maintainability, and extensibility.
- **Extensive Library Support:** C# comes with a rich standard library. This library can be used to perform a variety of tasks, such as file processing, network programming, database access, GUI development and more. Also, the .NET platform has a large ecosystem of third-party libraries and tools.
- **Advanced Language Features:** C# is constantly evolving as a modern programming language. Features added in recent years include `async/await`, LINQ (Language Integrated Query), nullable reference types. These features make code more readable, secure, and performant.
- **Comprehensive Development Tools:** Comprehensive integrated development environments (IDEs) such as Microsoft Visual Studio are available for C# development. These IDEs facilitate development processes such as writing, debugging, testing, and deploying code.

As seen in the

Fig, C# has become one of the most popular programming languages among the most popular programming languages.

Areas of Use

C# is a powerful programming language with broad industrial and academic support. Its high performance, reliability, and broad library support allow developers to build complex and scalable applications on a variety of platforms.¹⁷

- **Web Development and Web Services:** C# is used in conjunction with technologies such as ASP.NET and ASP.NET Core to develop web applications and websites. These technologies offer a powerful infrastructure and extensive library support.
- **Mobile Application Development:** C# is used in conjunction with frameworks such as Xamarin to develop mobile applications. This enables app development for Android and iOS platforms in a single code base.
- **Game Development:** C# is used in conjunction with the Unity game engine for game development. Since C# is

powerful, the famous game development platform, Unity, is preferred by many indie and professional game studios.

- **Desktop and Console Applications:** C# is used in conjunction with technologies such as Windows Presentation Foundation (WPF) and Windows Forms to develop desktop and console applications. This enables the creation of user-interface-oriented applications for the Windows operating system.
- **Data Science and Analytics:** C# is used to perform various data science and analytics operations on the .NET platform. In particular, programs written in C# can be used to process and analyze large data sets.
- **Financial Applications:** C# is widely used in the development of financial applications. In particular, it enables the creation of reliable and performant applications in the banking and finance sectors.
- **Software Agents and Systems:** C# is used in the development of automation tools, system tools, management tools, and other software tools. This enables the creation of easy-to-use and powerful tools in many different industries and business areas.

Advantages and Disadvantages Compared to Other Programming Languages

Advantages

- Integration with .NET Platform
- Object Oriented Programming (OOP) Support
- Extensive Library Support
- Advanced IDE Support

Disadvantages:

- Platform Dependency
- Learning Curve
- Strict Dependency
- Performance Issues

TEST ENVIRONMENT AND METADODOLOGY

Data Preparation

The data that to be loaded into the database with SQL DML operations is prepared in accordance with the following guidelines:

- The data was saved in JSON format as 20,000 randomly generated person data with 10 columns including ItemID, ItemGuid, Ad, Soyad, Doğum Tarihi, Adres, Telefon, LisansID, Numara, Özgeçmiş.
- Columns are created in int, uniqueidentifier, datetime, decimal, and nvarchar data types, which are commonly used in every database.
- The data is generated using a library called Faker, which has both a C# and Python library for generating random data.¹⁸ The library and the same data were taken from the same file and used in both programs.

The generation of the data was done with the following code snippets.

```

1. List<Kisi> kisiler;
2. public void Olustur()
3. {
4.     string sayi, path = string.Empty;
5.     Console.WriteLine("Lütfen oluşturulacak kişi sayısı girin.");
6.     sayi = Console.ReadLine();
7.     int ksayi;
8.     if (int.TryParse(sayi, out ksayi))
9.     {
10.        Console.WriteLine("Lütfen kişilerin kayıt edileceği dosya yolunu girin.");
11.        path = Console.ReadLine();
12.        int i = 1;
13.        kisiler = new List<Kisi>();
14.        while (i <= ksayi)
15.        {
16.            Kisi data = new Kisi
17.            {
18.                ItemID = i,
19.                ItemGuid = Guid.NewGuid(),
20.                Ad = Faker.Boolean.Random() == true ? Faker.Name.First() : Faker.Name.First() + " " + Faker.Name.Middle(),
21.                Soyad = Faker.Name.Last(),
22.                Adres = Faker.Address.StreetAddress(Faker.Boolean.Random()) + "/" + Faker.Address.City() + " - " + Faker.Address.Country(),
23.                DogumTarihi = Faker.Identification.DateOfBirth(),
24.                Numara = Faker.RandomNumber.Next(10000, 500000),
25.                LisansID = Faker.Identification.SocialSecurityNumber(),
26.                Ozgecmis = Faker.Lorem.Sentence(Faker.RandomNumber.Next(80, 150)),
27.                Telefon = Faker.Phone.Number()
28.            };
29.            kisiler.Add(data);
30.            i++;
31.        }
32.        JSONOlustur(kisiler, path);
33.    }
34.    Console.WriteLine(string.Format("{0} adet kişi, {1} dosyasına başarıyla kaydedildi. Çıkmak için Enter...",sayi,path));
35.    Console.ReadKey();
36. }

```

Code Snippet 1. The code that generates the test data

```

1. public class Kisi
2. {
3.     public int ItemID { get; set; }
4.     public Guid ItemGuid { get; set; }
5.     public string Ad { get; set; }
6.     public string Soyad { get; set; }
7.     public DateTime DogumTarihi { get; set; }
8.     public string Adres { get; set; }
9.     public string Telefon { get; set; }
10.    public string LisansID { get; set; }
11.    public decimal Numara { get; set; }
12.    public string Ozgecmis { get; set; }
13.
14.    public override string ToString()
15.    {
16.        return string.Format("ItemID:{0}\nItemGuid:{1}\nAd:{2}\nSoyad:{3}\n Adres:{4}\nDoğum Tarihi:{5}\nTelefon:{6}\n
LisansID:{7}\nNumara:{8}\n Ozgeçmiş:{9}", ItemID, ItemGuid, Ad, Soyad, Adres,DogumTarihi.ToShortDateString()
,Telefon,LisansID,Numara,Ozgecmis );
17.    }
18. }

```

Code Snippet 2. Person class code

An example of person data generated using Code Snippet 1 is shown below in JSON format.

```

1. {
2.   "ItemID": 1,
3.   "ItemGuid": "3a3b5691-f464-43d8-a56b-c8b2bf88d402",
4.   "Ad": "Hassan",
5.   "Soyad": "Auer",
6.   "DogumTarihi": "1985-09-11T00:00:00Z",
7.   "Adres": "6603 Judd Avenue Apt. 289 / South Ken - Guinea",
8.   "Telefon": "(883)857-2727 x9877",
9.   "LisansID": "157-78-1606",
10.  "Numara": 353583.0,
11.  "Ozgecmis": "Doloremque suscipit dicta qui consectetur non assumenda quos molestias voluptatem adipisci aspernatur aliquid sint quia ab ipsum corporis quo consequatur dolores natus dolor quae et consequatur et earum consequatur delectus odit praesentium exercitationem quia sunt culpa aperiam molestias id modi aut labore exercitationem voluptas illum suscipit dolores dolor sapiente animi optio laborum earum aut dignissimos et saepe nihil ex deserunt quaerat nesciunt et est at totam odio omnibus et sed aut natus earum qui ad et esse rerum temporibus labore et nihil est velit odit nam non dolores eveniet deserunt et eligendi autem praesentium non laborum placeat ex consequatur asperiores odit quisquam sed reprehenderit atque fuga laborum aut doloribus sed sit voluptatem omnis et doloribus aut accusamus excepturi quidem totam in eaque omnis non non sit aut fugit asperiores veniam molestiae consequuntur aut blanditiis aut eaque deserunt cum eos saepe odit harum impedit."
12. },

```

Code Snippet 3. Sample person data in JSON format

A total of 20,000 non-real-person data was created using Code Snippet 1 and this data was saved as a file in JSON format. This file was read into the prepared C# and Python programs and uploaded into the database, and tests were performed.

Preparation of Database

The database to perform SQL DML operations was created using

Code Snippet 4 created from on the SQL Server 2014 program installed on the same computer where the test programs run. The database was named KisilerVT, and has a single table named TB_Kisiler. The newly created database file is 5.00 MB (5,242,880 bytes), and the log file is 2.00 MB (2,097,152 bytes).

```

1. USE [master]
2. GO
3. CREATE DATABASE [KisilerVT]
4. ON PRIMARY
5. ( NAME = N'KisilerVT', FILENAME = N'C:\Test\KisilerVT.mdf' , SIZE = 5120KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
6. LOG ON
7. ( NAME = N'KisilerVT_log', FILENAME = N'C:\Test\KisilerVT_log.ldf' , SIZE = 2048KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
8. GO

```

Code Snippet 4. SQL code for creating a KisilerVT database

```

1. USE [KisilerVT]
2. GO
3. SET ANSI_NULLS ON
4. GO
5. SET QUOTED_IDENTIFIER ON
6. GO
7. CREATE TABLE [dbo].[TB_Kisiler](
8.     [ItemID] [int] NOT NULL,
9.     [ItemGuid] [uniqueidentifier] NOT NULL,
10.    [Ad] [nvarchar](50) NOT NULL,
11.    [Soyad] [nvarchar](50) NOT NULL,
12.    [Adres] [nvarchar](255) NOT NULL,
13.    [DogumTarihi] [datetime] NOT NULL,
14.    [Telefon] [nvarchar](50) NOT NULL,
15.    [Numara] [decimal](6, 0) NOT NULL,
16.    [LisansID] [nvarchar](50) NOT NULL,
17.    [Ozgecmis] [nvarchar](max) NOT NULL
18. ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
19. GO

```

Code Snippet 5. SQL code to create TB_Kisiler table

To create the KisilerVT database, and the TB_Kisiler table; Code Snippet 4 and Code Snippet 5 were used. When the testing of a language is complete, the database and table were dropped and re-created for another run using these codes.

Determination of Benchmark Conditions and Scenarios

Comparison Conditions

The conditions for comparison are as follows:

- Both programs were run on the same computer and through the Console Application in order to provide the same conditions and to get rid of the advantages and disadvantages of the IDE.
- The same data file was used in both programs.
- The database and table structure were rebuilt in both programming languages prior to testing.
- SQL DML operations are performed in the same way and in the same way from a software point of view, the difference being the two programming languages and the amount of code that needs to be written.
- In each comparison scenario, the same process was repeated three times, and the average of these three times was taken as the processing time.

Comparison Scenarios

As known, SQL DML operations cover 4 types of operations: INSERT, SELECT, UPDATE and DELETE.² Comparison scenarios were defined for each function. Random ItemID values were randomly selected between 1 and 20000 (11680) and saved as a txt file to be used in both programs.

The comparison for the INSERT function was made by adding new records through the following 3 different scenarios. These are;

- Inserting 20000 records by sending a separate query for each record,
- Inserting 20000 records by sending a query for every 100 records,
- Inserting 20000 records by sending a query for every 1000 records.

It was determined as adding it to the database and transaction times were recorded.

The comparison for the SELECT function was done by calling the records through the following 3 different scenarios. These are:

- Calling 20000 records in a single query and adding them to the list structure as Person object,
- Calling 5000 records in the order in which they are in the TB_Kisiler table, that is, in a single query with the TOP structure and adding them to the list structure as a Person object,
- Calling 11680 records in a single query with the WHERE structure over pre-selected ItemIDs and adding them to the list structure as Person objects.

It was determined as calling from the database and transaction times were recorded.

The comparison for the UPDATE function was made by updating the "Ozgecmis" field with the longest data over 3 scenarios with a new Ozgecmis value defined with the same value in both

programs. These are;

- Updating the "Ozgecmis" field of 20000 records in a single query,
- Updating 5000 records in the TB_Kisiler table, that is, updating the "Ozgecmis" field in a single query with the TOP structure
- Updating the "Ozgecmis" field in a single query with the WHERE structure over pre-selected ItemIDs of 11680 records,

Updates were made in the database, and transaction times were recorded.

The comparison for the DELETE function was made by deleting records in the following three different scenarios. These are;

- Deleting 20000 records from the database in a single query,
- Deleting 5000 records from the database in the order they are in the TB_Kisiler table, that is, in a single query with the TOP structure,
- Deleting 11680 records from the database in a single query with WHERE structure over pre-selected ItemIDs,

Deletion was made from the database and transaction times were recorded.

Preparation of Python and C# Programs

Python Program Codes

The Python program was prepared as 2 different files. These are; main.py file containing the start function and main functions, and kisi.py file containing the Kisi class.

C# Program Codes


The C# program was prepared as 2 different files. These are; Program.cs file containing the start function and main functions, and Kisi.cs file containing the Kisi class.

COMPARISON OF PERFORMANCE AND CODE WRITING

Performance Comparison

C# Performance Tests and Results

After all tests, the database file size was 81.0 MB (84,934,656 bytes), and the log file size was 214 MB (224,526,336 bytes), even though there was no data in the database.



```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAs\Drive\YBS Tez\Tez Files\DML CSharp Debug>"D:\Mahmut\YESILYURT NAs\Drive\YBS Tez\Tez Files\DML CSharp Debug\DML CSharp.exe"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3:UPDATE, 4:DELETE
1
INSERT İşlem Türü ? 1: TEKİL, 2: 100 lü GURUP, 3: 1000 lü GURUP
3
SQL Server Veritabanına INSERT işlemleri bağlayıyor...
Toplam 20000 kayıt, 1000 lü guruplar halinde insert edildi. İşlem 363 ms sürdü
Program Kapatılsın mı? (E/H)
  
```

Fig 1. Insert operation C# program screenshot

Table 1. C# Program Insert operation times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	3976	811	363
2. Operation	3797	792	323
3. Operation	3802	789	340
Average	3858	797	342


```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug\DML CSharp.exe"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
2
Select İşlem Türü ? 1: TOP 5000, 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına SELECT İşlemleri başlıyor...
Veritabanından TOP 5000 kayıt seçildi ve Kisi nesnesi olarak listelendi. İşlem 69 ms sürdü.
Program Kapatılınsın mı? (E/H)

```

Fig 2. Select process C# program screenshot

Table 2. C# Program Select processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	97	884	284
2. Operation	69	885	264
3. Operation	75	893	280
Average	80	887	276

```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug\DML CSharp.exe"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
3
UPDATE İşlem Türü ? 1: TOP 5000, 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına UPDATE İşlemleri başlıyor...
Veritabanında TOP 5000 kayıt güncellendi. İşlem 66 ms sürdü.
Program Kapatılınsın mı? (E/H)

```

Fig 3. Update process C# program screenshot

Table 3. C# Program Update processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	66	7105	265
2. Operation	62	7128	269
3. Operation	63	7132	258
Average	64	7122	264

```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DML CSharp Debug\DML CSharp.exe"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
4
DELETE İşlem Türü ? 1: 100'er (TOP 100) , 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına DELETE İşlemleri başlıyor...
Tüm kayıtlar, 100'er li gruplar halinde silindi. İşlem 597 ms sürdü
Program Kapatılınsın mı? (E/H)

```

Fig 4. Delete process C# program screenshot

Table 4. C# Program Delete processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	597	7017	45
2. Operation	500	6950	42
3. Operation	496	6998	45
Average	531	6988	44

Python Performance Tests and Results

After all tests, the database file size was 94.0 MB (98,566,144 bytes), while the log file size was 247 MB (258,994,112 bytes), even though there was no data in the database.

```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\venv\Scripts\python.exe" "D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\main.py"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
1
INSERT İşlem Türü ? 1: TEKİL, 2: 100 LÜ GURUP, 3: 1000 Lİ GURUP
3
SQL Server Veritabanına INSERT İşlemleri başlıyor...
Toplam 20000 kayıt, 1000 lü gruplar halinde insert edildi. İşlem 436.56709999777377 ms sürdü.
Program Kapatılınsın mı ? (E/H)

```

Fig 5. Insert process Python program screenshot

Table 5. C# Program Select processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	3358	871	436
2. Operation	3139	879	443
3. Operation	3176	878	422
Average	3224	876	434

```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\venv\Scripts\python.exe" "D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\main.py"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
2
Select İşlem Türü ? 1: TOP 5000, 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına SELECT İşlemleri başlıyor...
Veritabanından TOP 5000 kayıt seçildi ve Kisi nesnesi olarak listelendi. İşlem 61.1676999978954 ms sürdü.
Program Kapatılınsın mı ? (E/H)

```

Fig 6. Select process Python program screenshot

Table 6. C# Program Select processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	61	935	237
2. Operation	56	922	235
3. Operation	57	917	240
Average	56	925	237

```

C:\WINDOWS\system32\cmd. x + v
D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython>"D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\venv\Scripts\python.exe" "D:\Mahmut\YESILYURT NAS\Drive\YBS Tez\Tez Files\DMLPython\main.py"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
3
Update İşlem Türü ? 1: TOP 5000, 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına UPDATE İşlemleri başlıyor...
Veritabanında TOP 5000 kayıt güncellendi. İşlem 58.63619999785442 ms sürdü.
Program Kapatılınsın mı ? (E/H)

```

Fig 7. Update process Python program screenshot

Table 7. C# Program Select processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	58	6851	290
2. Operation	55	7245	270
3. Operation	57	6886	224
Average	57	6994	261

```

C:\WINDOWS\system32\cmd. x
D:\Mahmut\VESILYURT_NAS\Drive\YBS_Tez\Tez_Files\DMLPython>D:\Mahmut\VESILYURT_NAS\Drive\YBS_Tez\Tez_Files\DMLPython\.venv\Scripts\python.exe "D:\Mahmut\VESILYURT_NAS\Drive\YBS_Tez\Tez_Files\DMLPython\main.py"
Lütfen Yapılacak İşlemi Seçiniz: 1: INSERT, 2: SELECT, 3: UPDATE, 4: DELETE
4
DELETE İşlem Türü ? 1: 100'er (TOP 100), 2: RASTGELE, 3: TÜMÜ
1
SQL Server Veritabanına DELETE İşlemi başlıyor...
Toplam 28999 kayıt, 100' er li gruplar halinde silindi. İşlem 476.3987999994424 ms sürdü.
Program Kapatılsın mı ? (E/H)

```

Fig 8. Delete process Python program screenshot

Table 8. C# Program Select processing times test results (ms)

	Scenario 1	Scenario 2	Scenario 3
1. Operation	538	6778	43
2. Operation	490	7038	44
3. Operation	476	6834	37
Average	501	6883	41

Process Performance Comparison

Table 9. Comparison of both programming languages on Average Processing times

Process	C# Average Transaction Duration (ms)	Python Average Transaction Duration (ms)	Difference Rate (%)	Conclusion
Insert - S1	3858	3224	19,66%	Python
Insert - S2	797	876	9,87%	C#
Insert - S3	342	434	26,80%	C#
Select - S1	80	58	38,51%	Python
Select - S2	887	925	4,21%	C#
Select - S3	276	237	16,29%	Python
Update - S1	64	57	12,35%	Python
Update - S2	7122	6994	1,83%	Python
Update - S3	264	261	1,02%	Python
Delete - S1	531	501	5,92%	Python
Delete - S2	6988	6883	1,53%	Python
Delete - S3	44	41	6,45%	Python

Code Writing Comparison

Comparison of Code Size and Readability

C# is a statically typed programming language, which means that when defining a variable, its type must be specified. For example, a variable is declared as `int a = 5;`. Python, on the other hand, is a dynamically typed language. Here, no need to specify the type of variables in advance. A variable can simply be defined like `a = 5;`. This flexibility gives Python an advantage in rapid prototyping and code writing. However, in terms of safe code writing, statically typed languages are safer. Also, on the code readability side, this makes code readability difficult, and in large projects, it can be difficult to understand the code when looking at older code. At this point, statically typed languages are more advantageous.

The Python program code has a total of 9,688 characters, 203 lines, 891 words, and at a size of 9.46 KB (9,688 bytes).

The C# program codes totaled 16,566 characters, 330 lines, 1770 words, and at a size of 16.1 KB (16,556 bytes).

Table 10. Comparison of both programming languages on code parameters

Parameter	C#	Python	Ratio	Conclusion
Character	16566	9688	71,0%	Python
Line	330	203	62,6%	Python
Word	1770	891	98,7%	Python
Size (KB)	16,1	9,46	70,2%	Python

Debugging and Optimization Comparison

Although C# and Python are different software languages, they both have powerful tools for debugging and code optimization. C# offers debugging and optimization tools such as Visual Studio Debugger, Debug and Trace output generation classes "Debug.WriteLine()" and "Trace.WriteLine()" functions, real-time debug and break-point support, exception management with try-cache blocks, and profiling tools. Python, on the other hand, offers powerful debugging and code optimization tools with its powerful standard library and specially designed plugins for debugging. Some of these include the `pdb` and `pdb++` debugger classes, real-time debug and break-point support, exception handling with try-cache blocks, the `pytest` and `unittest` classes, and the logging class.

CONCLUSION

In this study, two separate computer programs were written to perform SQL Server DML operations using C# and Python, and with these programs, the content of the JSON file containing 10 columns and 20,000 rows of data was read with the program, the data was first converted into a list of objects, and the basic DML operations were implemented with the programs written in both languages over the specified process scenarios and comparisons were made in terms of both performance measurements and code writing. For performance measurements, each operation was repeated three times, and the average of these was considered the processing time.

Evaluation in terms of processing performance: A total of four operations were performed with three different scenarios, and although it seems that the Python language performed better mathematically when looking at the processing times of these 12 operations, it was observed that both programming languages did not provide a significant superiority over the other.

Evaluation in terms of code size and readability: In terms of code readability, C# can be considered a more readable language in a general context, but in the context of the programs written for this study, there was no difference between both programming languages in terms of code readability. Both in the general context and in the context of the program written within the scope of this thesis in terms of code size Table 11 when looking at the results in 'Python has a clear advantage.

Evaluation in terms of debugging and optimization: Both languages have been shown to have a high level of debugging and code optimization capabilities, so the main factor that will

determine the choice will be fitness for purpose.

As a result, it has been determined that both languages have superior features compared to each other, and that there is no definite superiority of these two languages over each other in DML operations. The choice of language should be based on the requirements of the project, the ecosystem, and the skills of the team.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – M.Y.; Design – M.Y.; Supervision – Y.Z.A., Resources – K.K.; Data Collection and/or Processing – Y.Z.A., M.Y.; Analysis and/or Interpretation – Y.Z.A., M.Y.; Literature Search – M.Y.; Writing Manuscript – Y.Z.A., M.Y.; Critical Review – Y.Z.A., M.Y.

Conflict of Interest: The authors have no conflicts of interest to declare.

Financial Disclosure: This study received no funding.

REFERENCES

1. Stackoverflow.com. Stack Overflow Developer Survey 2022. Stack Overflow. <https://survey.stackoverflow.co/2022#most-popular-technologies-misc-tech>. Accessed May 11, 2024.
2. Setiyadi D. Database System Development Life Cycle (DSDLC) on System Libraries for Data Manipulation Language (DML) Using SQL Server 2008. *J Mantik*. 2021;5(2):1065-1066. <https://www.ejournal.iocscience.org/index.php/mantik/article/view/1448/1032>. Accessed May 1, 2024.
3. Hakim Lubis J, Muisa Zamzami E. Relational Database Reconstruction from SQL to Entity Relational Diagrams. *J Phys Conf Ser*. 2020;1566(1). doi:10.1088/1742-6596/1566/1/012072.
4. Codd EF. A Relational Model of Data for Large Shared Data Banks. *Commun ACM*. 1970;13(6):377-387. doi:10.1145/362384.362685.
5. Microsoft Learn. ACID Properties - Win32 apps | Microsoft Learn. <https://learn.microsoft.com/en-us/windows/win32/cos SDK/acid-properties>. Accessed May 11, 2024.
6. Taylor P. Most popular relational DBMS 2022 | Statista. <https://www.statista.com/statistics/1131568/worldwide-popularity-ranking-relational-database-management-systems/>. Accessed May 11, 2024.
7. Azure Microsoft. NoSQL Database - What is NoSQL? <https://azure.microsoft.com/tr-tr/resources/cloud-computing-dictionary/what-is-nosql-database/>. Accessed May 11, 2024.
8. Amazon.com. NoSQL Nedir? | Non-Relational Databases, Flexible Schema Data Models | AWS. <https://aws.amazon.com/tr/nosql/>. Accessed May 11, 2024.
9. Bektaş G. The Importance of Data Management and the Role of NoSQL Databases: ACID, CAP, and BASE Concepts and Principles. <https://medium.com/@bektasguner772/the-importance-of-data-management-and-the-role-of-nosql-databases-acid-cap-and-base-concepts-and-principles-e7c572742098>. Accessed May 11, 2024.
10. Innuy. List of Best and Most Popular NoSQL Database 2022. <https://www.innuy.com/blog/list-of-best-and-most-popular-nosql-database-2022/>. Accessed May 11, 2024.
11. Saralioğlu E. *Python Programlama Dili Kullanılarak Uzaktan Algılama Amaçlı Arayüz Tasarımı* [master's thesis]. Karadeniz Technical University; 2015.
12. Ayvaz U, Çoban A, Gürüler H, Peker M. Python Dilinin Öznitelikleri, Programlama Eğitiminde ve Yazılım Dünyasındaki Yeri. *Akad Bilişim*. 2016;1-6.
13. Summerfield M. *Programming in Python 3: A Complete Introduction to the Python Language*. 2nd ed. http://ifile.it/qx63lw/ebooksclub.org__Programming_in_Python_3__A_Complete_Introduction_to_the_Python_Language__2nd_Edition_.pdf. Accessed May 9, 2024.
14. Scarlett R. Why Python keeps growing, explained - The GitHub Blog. <https://github.blog/2023-03-02-why-python-keeps-growing-explained/>. Accessed May 11, 2024.
15. Python.org. Python 3 Documentation. <https://docs.python.org/>. Accessed May 11, 2024.
16. Kimmel P. *Advanced C# Programming*. McGraw-Hill/Osborne; 2002.
17. Microsoft Learn. C# Guide .NET managed language | Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/>. Accessed May 12, 2024.
18. Riches O. GitHub - oriches/faker-cs: C# port of the Ruby Faker gem. <https://github.com/oriches/faker-cs>. Accessed May 11, 2024.