



**Armen Melkumyan** • 1st  
Technical / Solutions Architect  
1yr •

...

## Implementing Asynchronous Operations in [ADO.NET](#)

Question:

How can you implement asynchronous database operations in [ADO.NET](#) to enhance application performance?

Answer:

Asynchronous operations in [ADO.NET](#) prevent blocking the calling thread and improve application responsiveness, especially for I/O-bound tasks like database operations.

Explanation:

Async Methods: `OpenAsync`, `ExecuteReaderAsync`, and `ReadAsync` are used for non-blocking operations.

`await` Keyword: Ensures the method asynchronously waits for the operation to complete without blocking the main thread.

[#csharp](#) [#dotnet](#) [#adonet](#) [#optimisation](#) [#adonet](#) [#asyncprogramming](#)

```
Implementing Asynchronous Operations in ADO.NET

1 public async Task<List<User>> GetAllUsersAsync()
2 {
3     var users = new List<User>();
4     string query = "SELECT * FROM Users";
5
6     using (SqlConnection conn = new SqlConnection(connectionString))
7     {
8         await conn.OpenAsync();
9         using (SqlCommand cmd = new SqlCommand(query, conn))
10        {
11            using (SqlDataReader reader = await cmd.ExecuteReaderAsync())
12            {
13                while (await reader.ReadAsync())
14                {
15                    users.Add(new User { /* Populate user data */ });
16                }
17            }
18        }
19    }
20    return users;
21 }
22
```

codetoint.com