In **Entity Framework Core**, you can perform **bulk deletes** in two main ways:

- **Using EF Core 7+ built-in** `ExecuteDelete` / `ExecuteDeleteAsync` (no extra packages needed).
- **Using third-party libraries** like EFCore.BulkExtensions or Z.EntityFramework.Extensions for older EF Core versions or advanced scenarios.

1️⃣ EF Core 7+ Built-in Bulk Delete

This is the most efficient and recommended way if you are on EF Core 7 or later.

```
using Microsoft.EntityFrameworkCore;
using System;
using System.Threading.Tasks;

public class MyDbContext : DbContext
{
    public DbSet<User> Users { get; set; }
}

public class User
{
    public int Id { get; set; }
    public string Name { get; set; }
}

class Program
{
```

```
    static async Task Main()
    {
        using var context = new MyDbContext();

        // Example: Delete all inactive users in one SQL statement
        int deletedCount = await context.Users
            .Where(u => !u.IsActive)
            .ExecuteDeleteAsync();

        Console.WriteLine($"{deletedCount} users deleted.");
    }
}
```

✅ **Advantages**:

- Executes a single DELETE SQL statement directly in the database.
- No entity tracking or loading into memory.
- Very fast for large datasets.

2️⃣ EFCore.BulkExtensions (For EF Core 3–6 or advanced needs)

If you are on EF Core < 7 or need more control, you can use **EFCore.BulkExtensions**.

Install via NuGet:

```
dotnet add package EFCore.BulkExtensions
```

Example:

```
using EFCore.BulkExtensions;
using System;
using System.Threading.Tasks;

class Program
{
    static async Task Main()
    {
        using var context = new MyDbContext();

        // Delete all inactive users
        await context.BulkDeleteAsync(
            context.Users.Where(u => !u.IsActive).ToList()
        );
    }
}
```

⚠ **Note**: This approach loads matching entities into memory before deleting, unless you use raw SQL or custom batch delete logic.

3️⃣ Raw SQL for Maximum Control

If you need full SQL control:

```
await context.Database.ExecuteSqlRawAsync(
    "DELETE FROM Users WHERE IsActive = 0"
);
```

⚡ Fastest possible, but you lose EF Core abstraction and compile-time safety.

✅ **Recommendation**:

- **EF Core 7+** → Use `ExecuteDeleteAsync()` (best performance, no extra package).
- **Older EF Core** → Use EFCore.BulkExtensions or raw SQL.

If you want, I can give you a **full working EF Core 7 bulk delete example with migrations and seeding** so you can run it immediately.
Do you want me to prepare that?