In **Entity Framework Core**, when you're dealing with related entities (like foreign key relationships), **how and when** those related entities are loaded from the database matters a lot. This is where **Eager**, **Lazy**, and **Explicit** loading come in.

**🏗️ Let's first understand the context**

Assume you have two entities:

public class Author

{

public int Id { get; set; }

public string Name { get; set; }

// Navigation property

public List<Book> Books { get; set; }

}

public class Book

{

public int Id { get; set; }

public string Title { get; set; }

// Foreign key

public int AuthorId { get; set; }

public Author Author { get; set; }

}

**🔍 1. Eager Loading**

**✅ What is it?**

Loads related data **at the same time** as the main entity — in a **single query** using Include().

**📌 Example:**

var authors = context.Authors

.Include(a => a.Books)

.ToList();

🔹 This retrieves **authors and their books** together in **one SQL query** using a JOIN.

**✅ When to use?**

* You know you'll need related data immediately.
* You want to reduce database round-trips.

**🐢 2. Lazy Loading (Needs setup in EF Core)**

**✅ What is it?**

Loads related data **only when you access the navigation property** — triggers a new SQL query behind the scenes.

**⚠️ EF Core does NOT support Lazy Loading by default. You must:**

* Install package: Microsoft.EntityFrameworkCore.Proxies
* Enable proxy creation
* Make navigation properties **virtual**

**📌 Setup Example:**

**Install:**

dotnet add package Microsoft.EntityFrameworkCore.Proxies

**Enable in DbContext:**

options.UseLazyLoadingProxies();

**Update model:**

public virtual List<Book> Books { get; set; }

**Usage:**

var author = context.Authors.First(); // books are not loaded yet

var books = author.Books; // books are loaded \*now\* via a new SQL query

**✅ When to use?**

* You want to load related data **only when needed**
* You’re okay with **multiple small queries**

**🖐️ 3. Explicit Loading**

**✅ What is it?**

You load related data **manually** after the main entity is loaded.

**📌 Example:**

var author = context.Authors.First();

context.Entry(author)

.Collection(a => a.Books)

.Load(); // loads books only for this author

Or for reference navigation:

context.Entry(book)

.Reference(b => b.Author)

.Load();

**✅ When to use?**

* You don’t want Lazy Loading
* You want full control over when to load related data

**🔁 Summary Table**

| **Type** | **Trigger Time** | **Number of Queries** | **Requires Configuration** |
| --- | --- | --- | --- |
| **Eager Loading** | Immediately (with query) | 1 | ❌ No |
| **Lazy Loading** | On property access | Multiple | ✅ Yes (Proxies + virtual) |
| **Explicit Loading** | Manually controlled | 2+ | ❌ No |

**🧠 Tip**

Use **Eager loading** when you need the related data for a list/grid view.  
Use **Lazy/Explicit loading** when you need related data **only in specific situations** to reduce unnecessary data fetching.