|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Framework Core** | | | | | | |
| **Title** | **Define** | **SubTitle** | **Code** | **Code Details** | **Note** | **Image** | |
| **Benefits of Repository Pattern** |  **Abstraction**: Separates data access logic from business logic, making it easier to change the data source without affecting the business layer.   **Testability**: Enables unit testing by allowing mocking of repositories.   **Encapsulation**: Centralizes data access logic, preventing duplication.   **Maintainability**: Makes it easier to modify and extend the data access layer.   **Decoupling**: Reduces direct dependency on ORM (like Entity Framework) in business logic.   **Consistency**: Standardizes how data operations are performed across the application. |  |  |  |  The **Repository Pattern** abstracts the data access logic.     **Unit of Work** ensures that multiple database operations are committed in a single transaction.   | This approach **improves maintainability, testability, and separation of concerns** in .NET applications. | |