|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Technique Name | Definition | Advantage | Usage | Disadvantages |
| ADO (NO ORM) | Techniques to build connection with db with fully depends on sql command and query in order to reflect theme and execute in db side the get the result back inside the C# solution | 1. Full Control on Db and Query and schema object 2. High Performance 3. Flexibility | 1. Big System 2. Complex Relation System 3. Data Analytic Systems 4. Reporting System | 1. Make Code unreadable 2. Less Level of Security 3. Logic Migration From DBMS to another 4. No Direct Mapping for Db Result |
| EF / NHibernate | Full ORM designed to build integration with db with out use SQL command / query by replace sql commad / query with LINQ Queries and replace Tables with Object and Also Handle / Map Query Result into Object | 1. Usability 2. Support Db First & Code First 3. Scalable and Maintainable 4. Querable / Memoery Execuation approach | 1. Small / Mid System 2. Less Complex Relational System | 1. Bad Performance Scenarios |
| Dapper (Micro ORM) | Techniques to build connection with db with minimum usage of sql command and query  and use C# object to handel result and response from db | 1. Light weight 2. High performance 3. Flexibility to handle largest amount of business case | 1. Mid / Small App |  |

ال ADO مع البروكجت الكبير والي بدك فيه اداء عالي وتحكم مطلق في مكونات الداتا بيس

ال Dapper مع البروجكتات المتوسطه او الصغيره والي بدك فيها سهوله استخدام مع سرعه

ال EF مشاريع متوسطه وصغيره وبدها سهوله تطوير واداء مقبول   
ال NHibernate للتطبيقات المعقده والكبيره والي بحاجه مرونه عاليه واداء ممتاز

ORM : Object Relation Mapper/Mapping

Layer To Integrate C# Applications With Data base By Reducing the Gap Between C# Code and SQL Syntax

Using Objects (Reduce SQL Command and Code Usage) To Let Me Code Readable and Clear and Maintabale and Secure

It's Mainly stand to convert Table / Result set (view , sp , schema obj) to C# Class and Via Versa (Tables)

Types

1- Default ( No Any Using For SQL Command ) (EF - Nhibernate)

2- Micro ORM (usage for sql command / query ) --> Hyprid between ADO and Other ORM ----> (Dapper)

Advantages

1- Reduce Syntax Error in SQL Commad

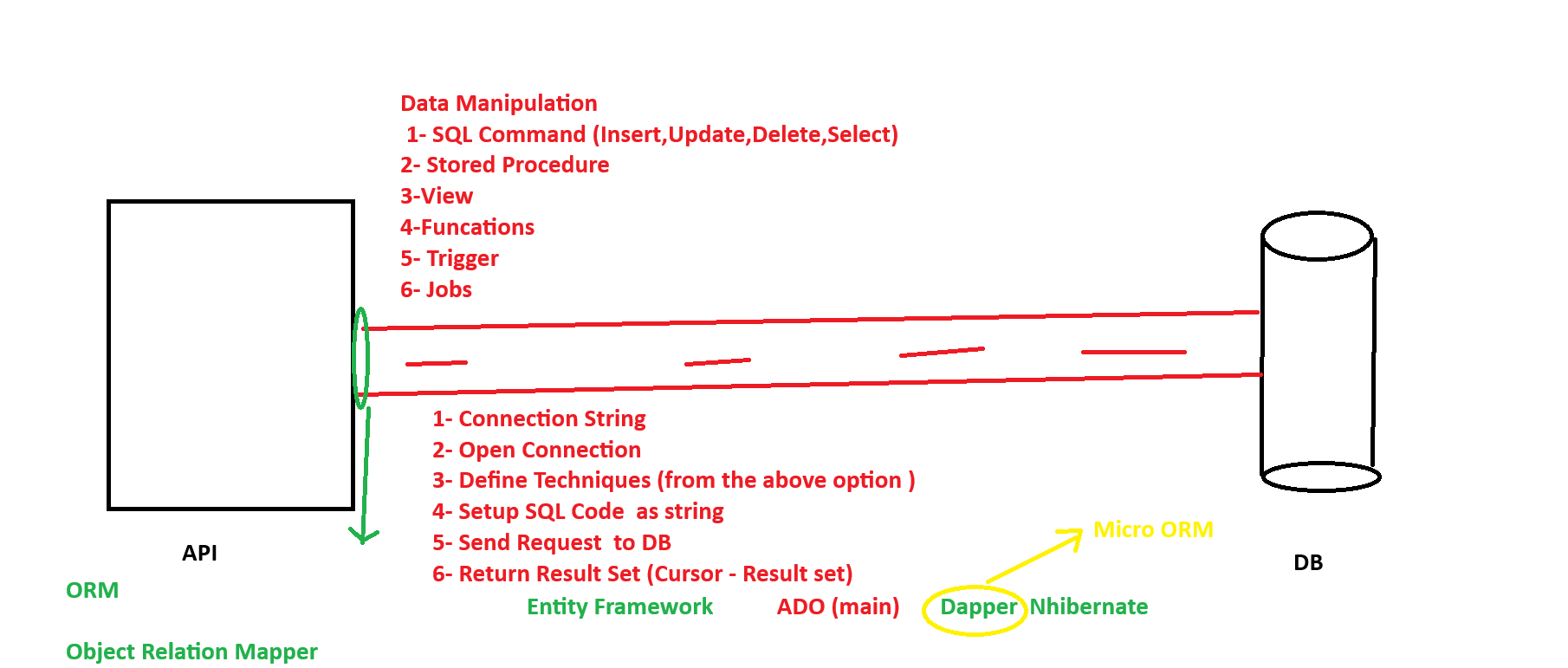
2- Handling Relationships

3- Less Code and Reduce Development Time (Produactivity)

4- Type Safe

5- Less Dependacies on DBMS

6- Secure

7- easy to scale(scalability) Proccessing inside Endpoints

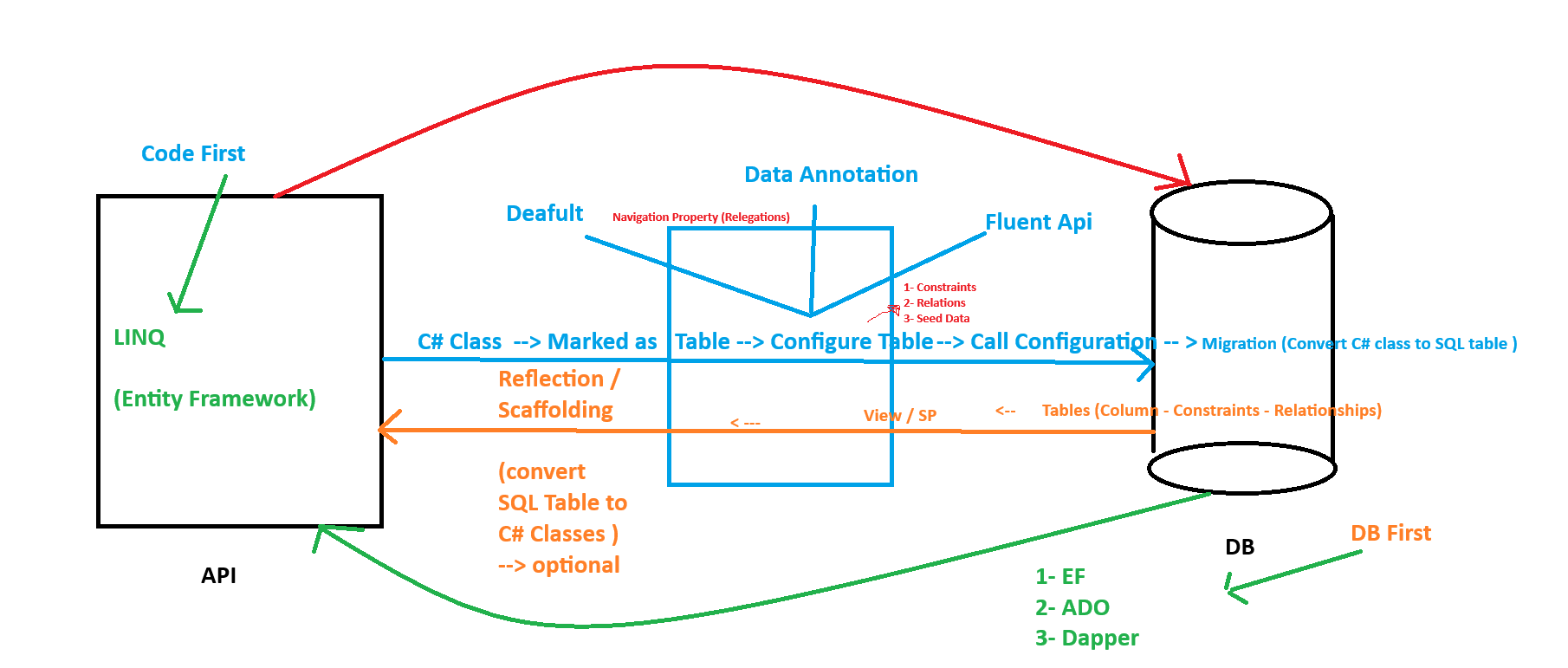
1- Validate input parameter

2- setup external resource

3- apply integration

4- apply logic

5- preparing responses (compatible datatype / nullable check) Seteps to Work With ADO.NET



1- Establish Connecation String

2- Download One Package Depends on the used DBMS

Download One Package To build Generic Connection With Different DBMS -/

Microsoft.Data.SQLClient ---> Latest Version

Command ---> Install-Package Micorosoft.Data.SQLClient (Nuget Package Manager Console)

UI - Element --> (Nuget Package Manager)

a- select Tools Tabl

b- Move / Select Nuget Package Manager Option

3-

if (string.IsNullOrWhiteSpace(Email))

throw new Exception("Email and Password are required");

SqlConnection sqlConnection = new SqlConnection(\_connectionString);

SqlCommand command = new SqlCommand("GenerateOTP", sqlConnection);

command.CommandType = System.Data.CommandType.StoredProcedure;

command.Parameters.AddWithValue("@Email", Email);

await sqlConnection.OpenAsync();

var resulte = await command.ExecuteNonQueryAsync();

return Ok(new

{

OTP = resulte,

});

/////////////////////////

//create connection

cnn=new SqlConnection

(Database.GetConnectionString());

cmd=new SqlCommand();

//open connection

cnn.Open();

//set command properties

cmd.Connection=cnn;

cmd.CommandText="select \* from employees";

//get query results in data reader

dr=cmd.ExecuteReader();

//loop through reader and output values

while(dr.Read())