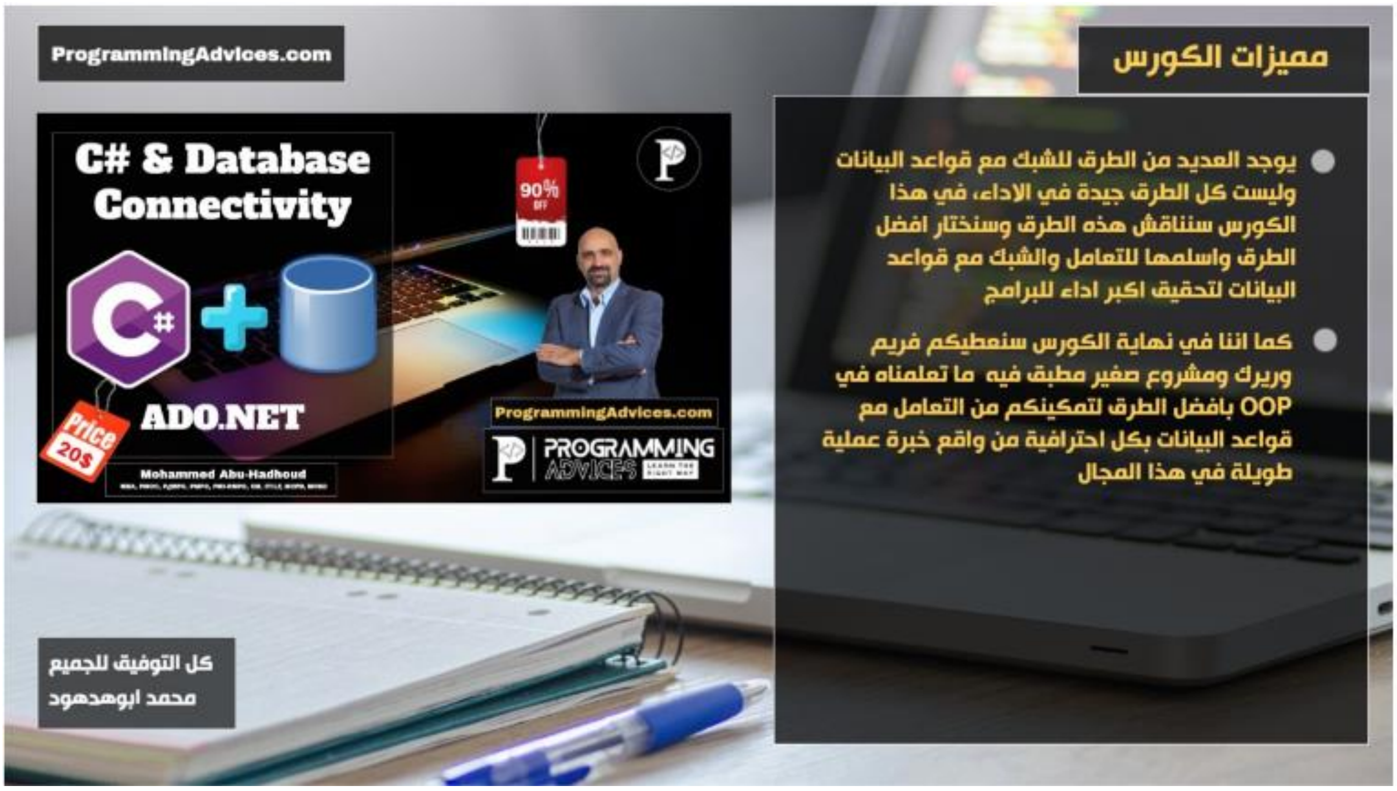
About This Course

هنركز علي حتة اننا ازاي نربط قاعدة البيانات مع ال c#

فيه طرق مختلفه عشان تربط مع الداتا بيز فيه طرق سهله وطرق صعبه

لو اخترت طريقه سهلة ال performance بتاع التطبيق بتاعك هيكون ابطأ

هنا عاوزك قبل ماتبدأ تفكر في أي حاجه تفتكر مشروع البنك وانك عندك المقدره انك تتعامل مع الكورس ده بسهوله لان العقبه اللي بتحول بينك وبين الكورس هي انك تجيب الداتا او تبعتها للداتا بيز بس كده والباقي هيكون هوا نفسه اللي كنت بتعمله في تطبيق البنك بس بدل ماكنت بتعمل العمليات علي ال text file لا هتعملها علي الداتا بيز والموضوع هنا ديته سطرين كود هتجيب الداتا متفلتره ومتظبطه ناقص بس انك تخزنها في structure مش اكتر


ProgrammingAdvices.com
C# & Database Connectivity
ADO.NET
90% OFF
Price 20$
Mohammed Abu-Hadhoud
ProgrammingAdvices.com
PROGRAMMING ADVICES
كل التوفيق للجميع
محمد ابوهدهود

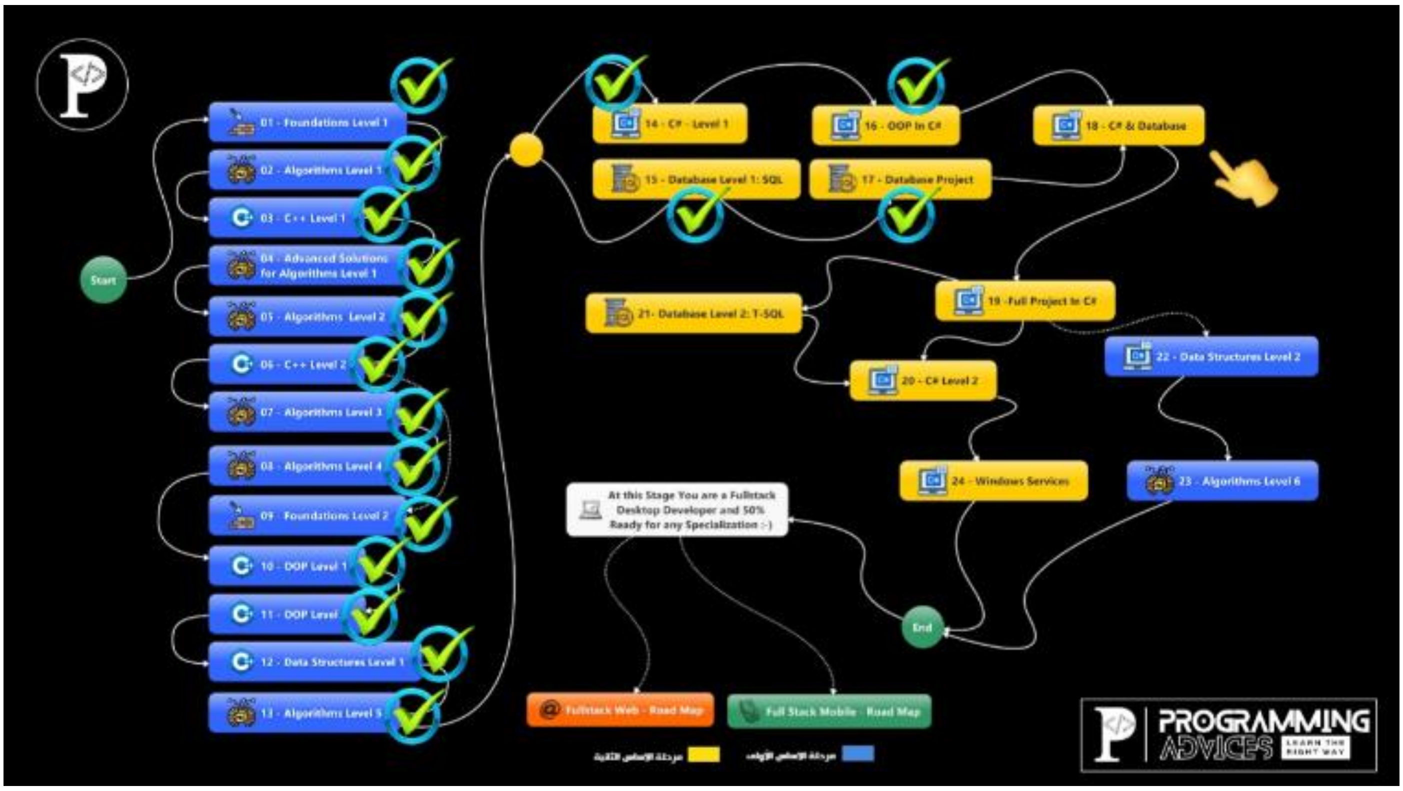
مميزات الكورس

- يوجد العديد من الطرق للشبك مع قواعد البيانات وليست كل الطرق جيدة في الاداء، في هذا الكورس سنناقش هذه الطرق وسنختار افضل الطرق واسلمها للتعامل والشبك مع قواعد البيانات لتحقيق اكبر اداء للبرامج
- كما اننا في نهاية الكورس سنعطيكم فريم وريرك ومشروع صغير مطبق فيه ما تعلمناه في OOP بافضل الطرق لتمكينكم من التعامل مع قواعد البيانات بكل احترافية من واقع خبرة عملية طويلة في هذا المجال

مهم جدا

يجب ان تكون قد انهيت جميع الكورسات المشار اليها بالاخضر

تستطيع ان تبدأ هذا الكورس جنبا الى جنب مع كورس 17



Telegram Group for This Course

رابط المجموعة على التلجرام

الرجاء عدم مشاركة هذا الرابط مع احد

https://t.me/+0EIzap2sB4lmZWY8

https://t.me/+0EIzap2sB4lmZWY8

What is XML?

https://www.youtube.com/watch?v=3WLKXzTCWEs

مراجعه علي درس ال xml وانه اختصار ل extensible markup language وهوا عباره عن ملف نصي بيتم تخزين فيه الداتا ونقلها بحيث انه يكون من السهل الوصول ليها عن طريق البرنامج وسهل علي الانسان يقراها

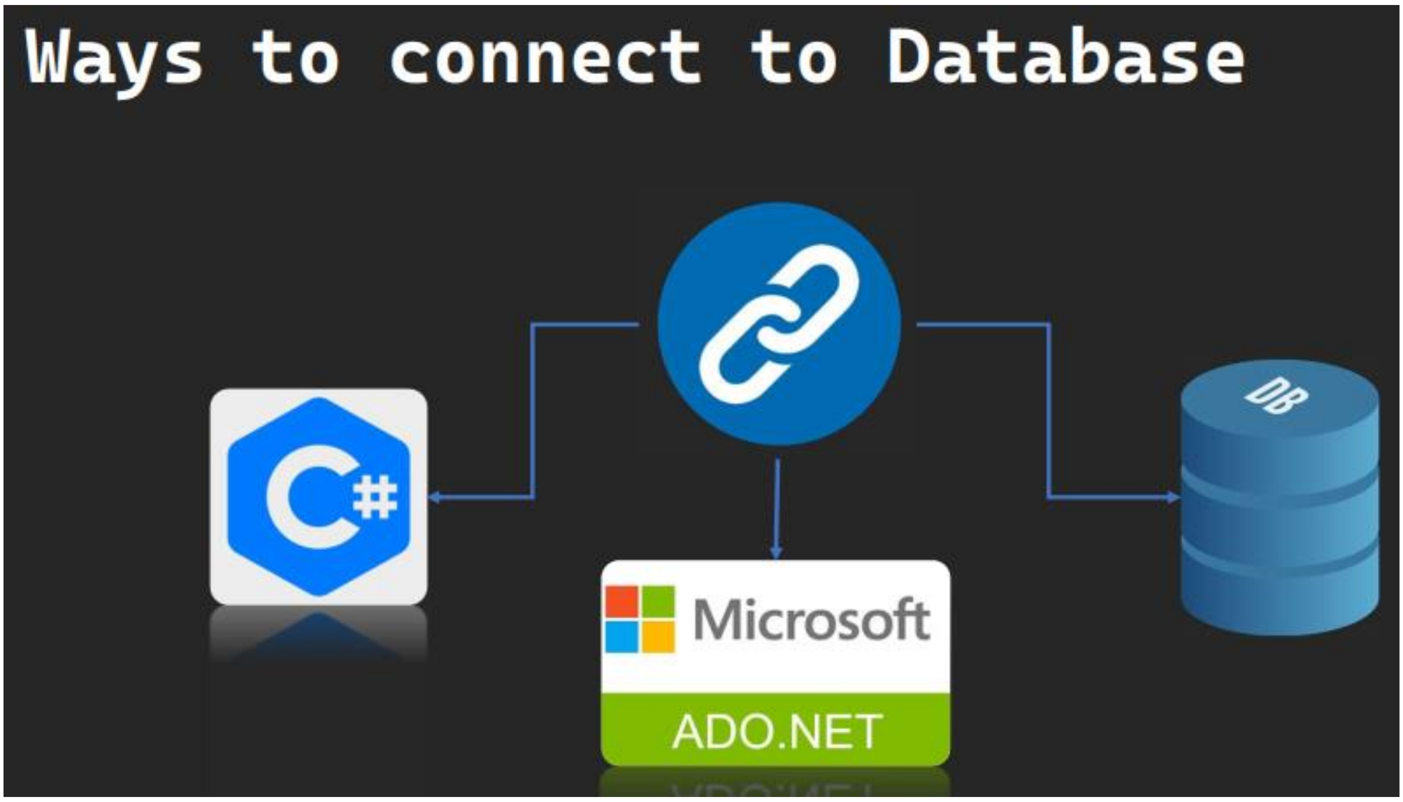
What is ADO.NET?

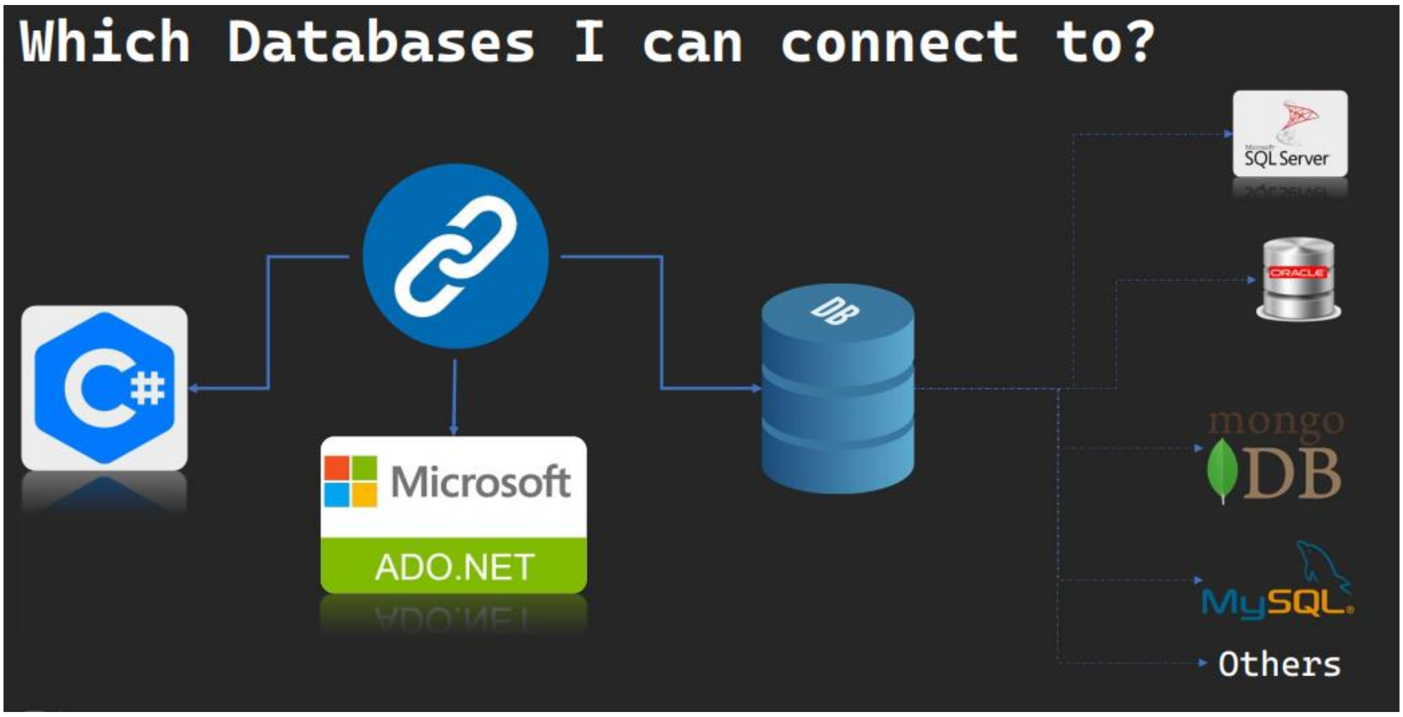
هنتعلم اننا نربط الداتا بيز بالبرنامج عن طريقه احدي الطرق الصعبه بس هتكون افضل للبرنامج

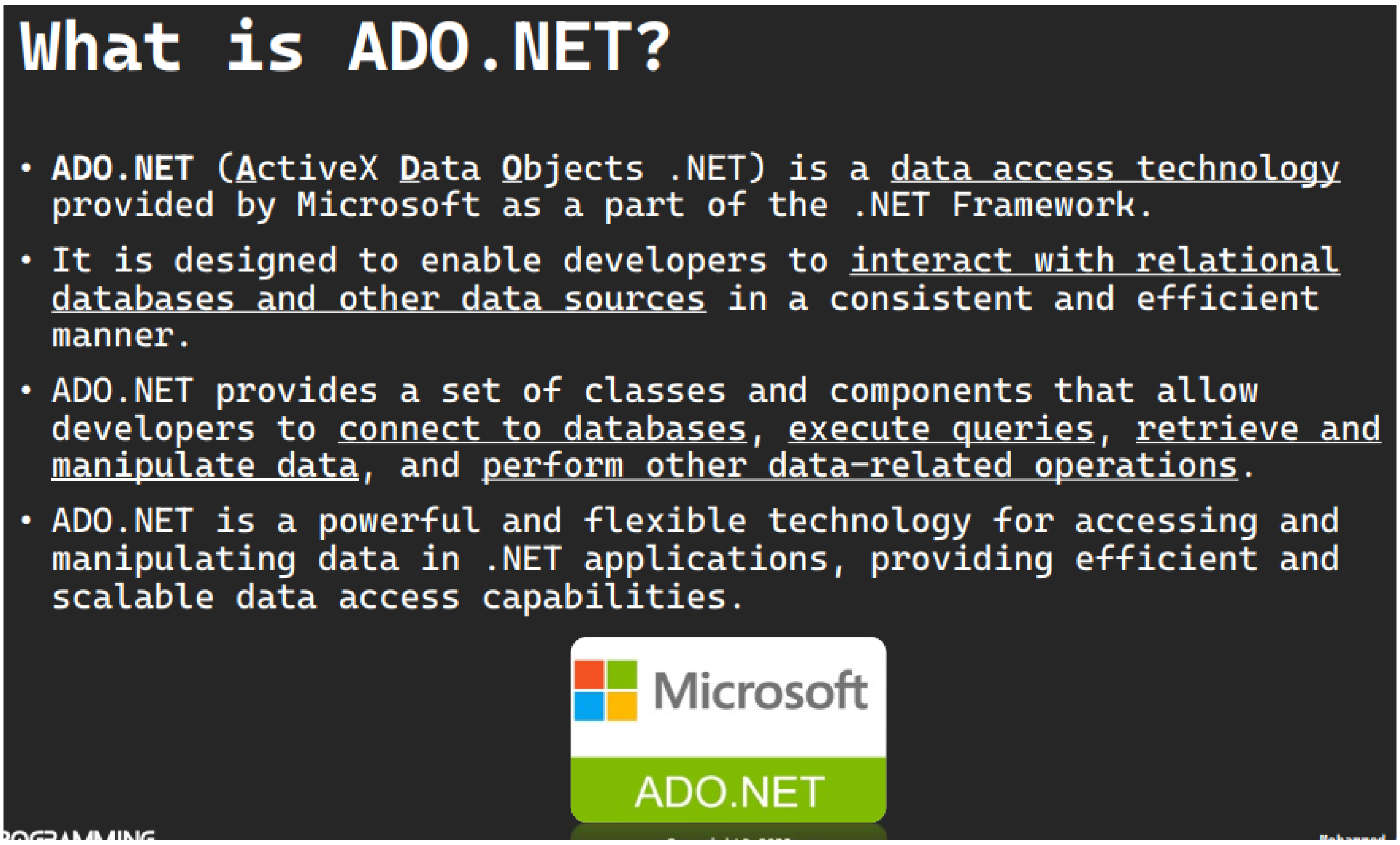
طيب بالنسبالنا هنا عشان نربط ال c# بالداتا بيز فيه طريقه واحده بس اسمها ADO.NET وهيا اختصار ل ACTIVE X DATA OBJECT وهيا بتخليك تنفذ أوامر علي الداتا بيز

فيه حاجه تانيه اسمها entity framework وده عباره عن مكتبات مبنيه علي ال ado.net بتريحك شوية في الشغل لكنها ابطأ

ال ado.net بتخليك تقدر تبرط البرنامج مع أنواع كتير من الداتا بيز زي ال sql server وال oracle database و mongo database وال my sql وغيره







What is ADO.NET?

- **ADO.NET** (**A**ctiveX **D**ata **O**bjects .NET) is a data access technology provided by Microsoft as a part of the .NET Framework.
- It is designed to enable developers to interact with relational databases and other data sources in a consistent and efficient manner.
- ADO.NET provides a set of classes and components that allow developers to connect to databases, execute queries, retrieve and manipulate data, and perform other data-related operations.
- ADO.NET is a powerful and flexible technology for accessing and manipulating data in .NET applications, providing efficient and scalable data access capabilities.

What is ADO.NET?

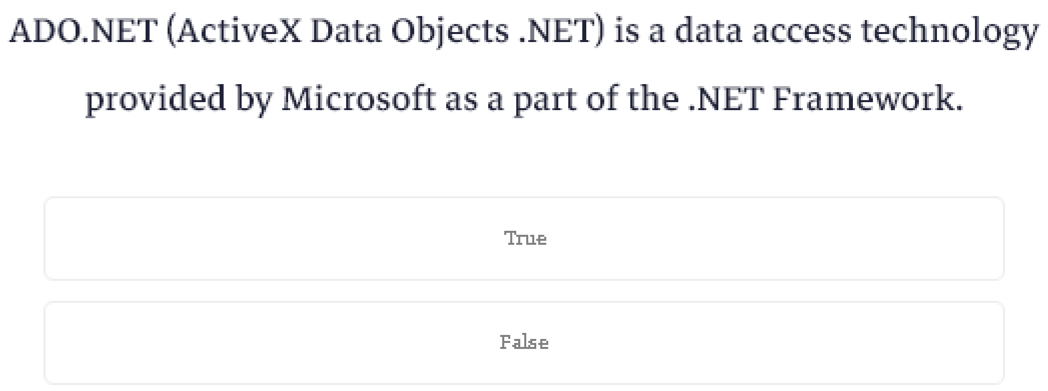
ADO.NET is a module of .Net Framework which is used to establish connection between application and data sources. Data sources can be such as SQL Server and XML. ADO.NET consists of classes that can be used to connect, retrieve, insert and delete data.

What types of Applications use ADO.NET?

ADO.NET can be used to develop any kind of .NET application. The following are some of the .NET applications where you can use ADO.NET Data Access Technology to interact with a data source.

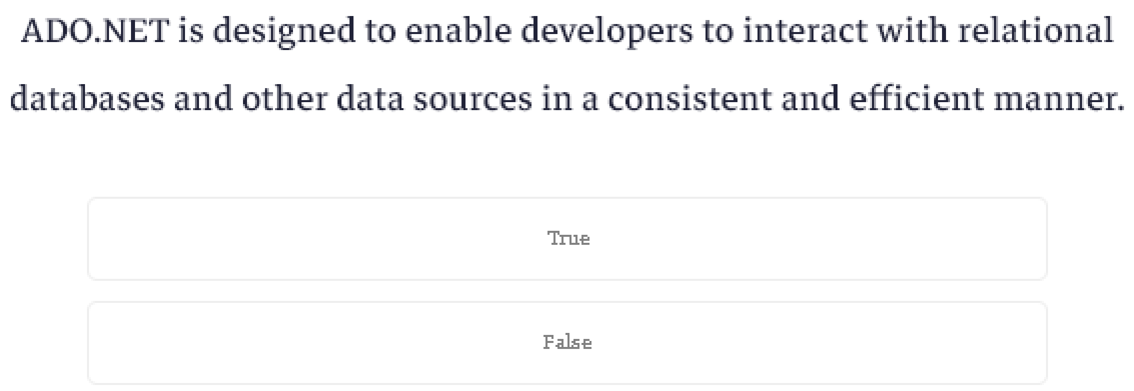
1. **ASP.NET Web Form Applications**
2. **Windows Applications**
3. **ASP.NET MVC Application**
4. **Console Applications**
5. **ASP.NET Web API Applications**
6. **ASP.NET Core Applications**

الواجب


ADO.NET (ActiveX Data Objects .NET) is a data access technology provided by Microsoft as a part of the .NET Framework.

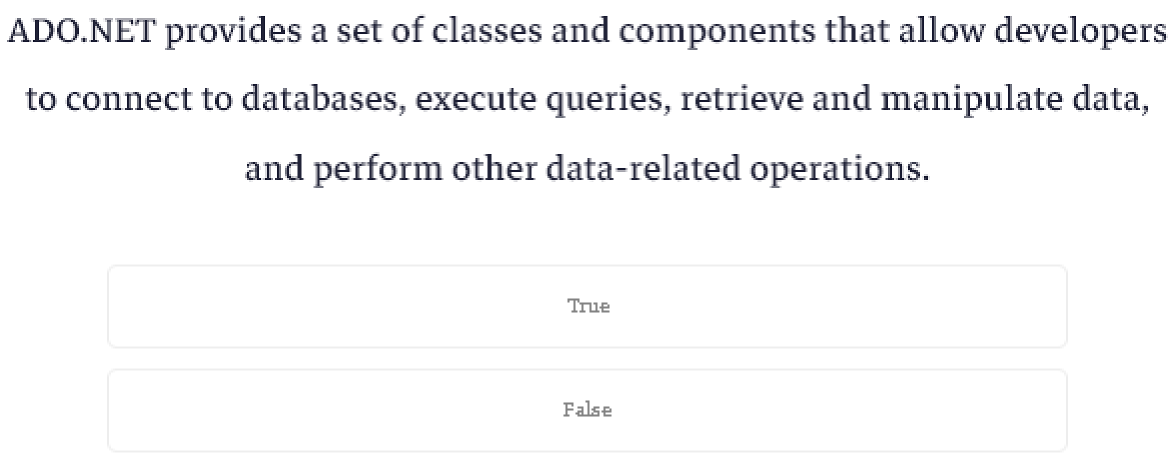
True

False


ADO.NET is designed to enable developers to interact with relational databases and other data sources in a consistent and efficient manner.

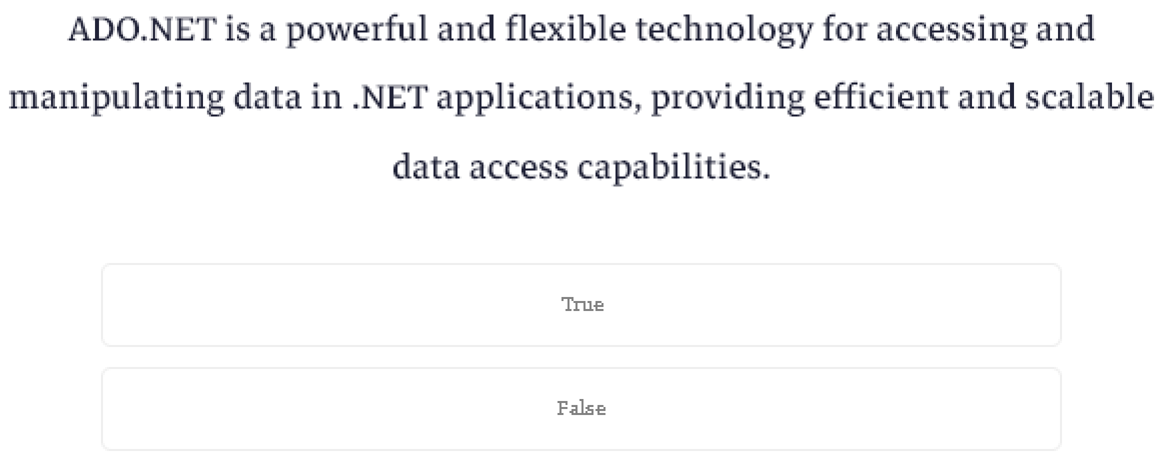
True

False


ADO.NET provides a set of classes and components that allow developers to connect to databases, execute queries, retrieve and manipulate data, and perform other data-related operations.

True

False


ADO.NET is a powerful and flexible technology for accessing and manipulating data in .NET applications, providing efficient and scalable data access capabilities.

True

False

ADO.NET Framework Data Providers

ال ado.net بيخلي البرنامج يتوصل مع الداتا بيز عن طريق ال data provider

ودي بتكون عباره عن مكتبات

زي مثلا مكتبه اسمها system.data.sqlclient ودي وظيفتها انها تربط ال sql server ودي اسرع طريقه عشان تربط بال sql server فيه حاجات تانيه بس دي الأسرع

عشان تربط مع ال oracle database بتستخدم ال system.data.oracleclient

System.data.mysqlclient ودي عشان تربط مع ال mysql

Mongodb.driver عشان تربط مع ال mongo db

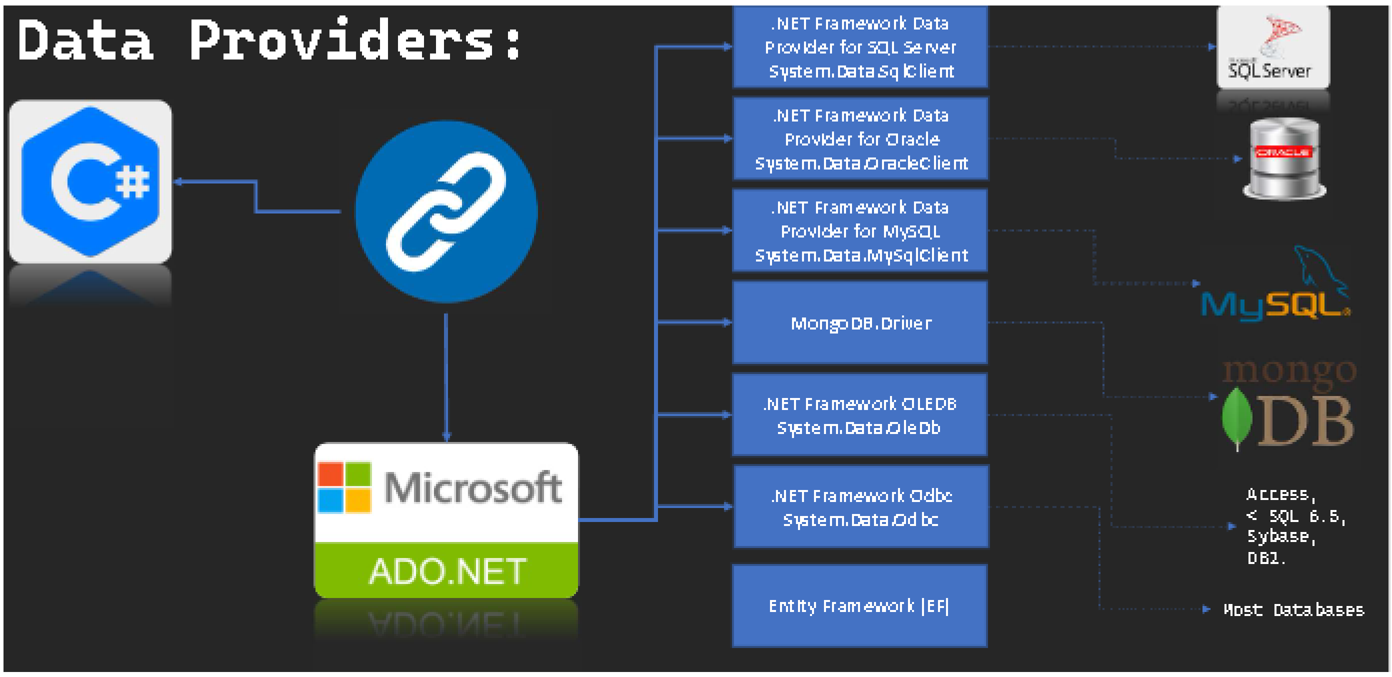
فيه ال system.data.oledb ودي بتخليك تتواصل مع كل أنواع الداتابيز بما فيهم اللي فوق وهيا اختصار ل object link embedded database وهيا general data provider وهيا معموله للويندوز بس الأفضل انك تستخدم اللي فوق

فيه حاجه تانيه معموله لكل ال operating systems واسمها odbc

System.data.odbc

وزي ماقولنا فيه عندك ال entity frame work وهيا مبنيه علي ال entity framework

الداتا تقدر ترجعها وتخليها read only او انك ترجعها في حاجه اسمها disconnected mode عن طريق انك تعملها caching يعني تحملها عال cache memory


Data Providers:
C#
Microsoft
ADO.NET
.NET Framework Data Provider for SQL Server System.Data.SqlClient
.NET Framework Data Provider for Oracle System.Data.OracleClient
.NET Framework Data Provider for MySQL System.Data.MySqlClient
MongoDB.Driver
.NET Framework OLEDB System.Data.OleDb
.NET Framework Odbc System.Data.Odbc
Entity Framework (EF)
SQL Server
ORACLE
MySQL
mongo DB
Access, < SQL 6.5, Sybase, DB2.
Most Databases

What is Data Provider?

.NET Framework data provider is used for:

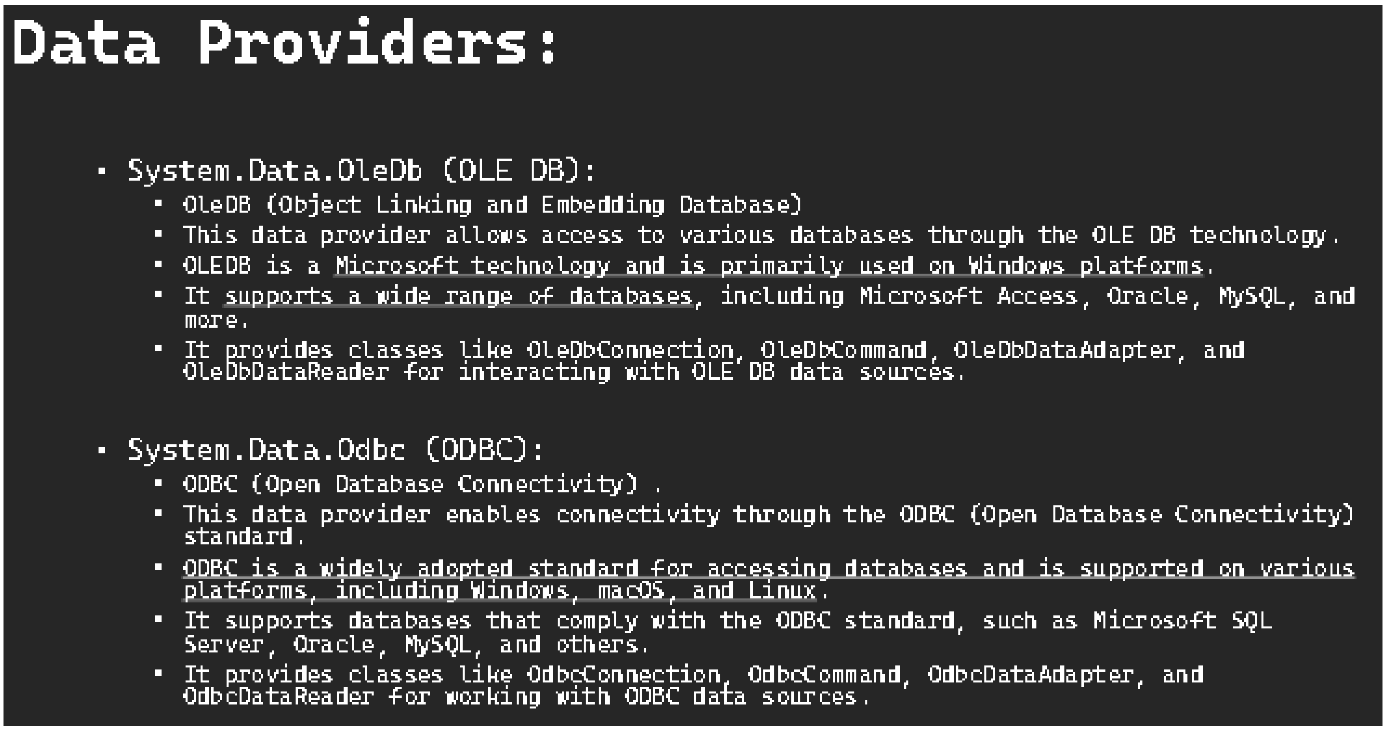
- Connecting to a database
- Executing commands,
- And retrieving results.

Those results are either processed directly, placed in a DataSet in order to be exposed to the user as needed, combined with data from multiple sources, or remoted between tiers.

ADO.NET supports various data providers, including SQL Server, Oracle, MySQL, and OLE DB. It provides a consistent programming model for working with different databases, allowing developers to write data access code that is independent of the underlying database.

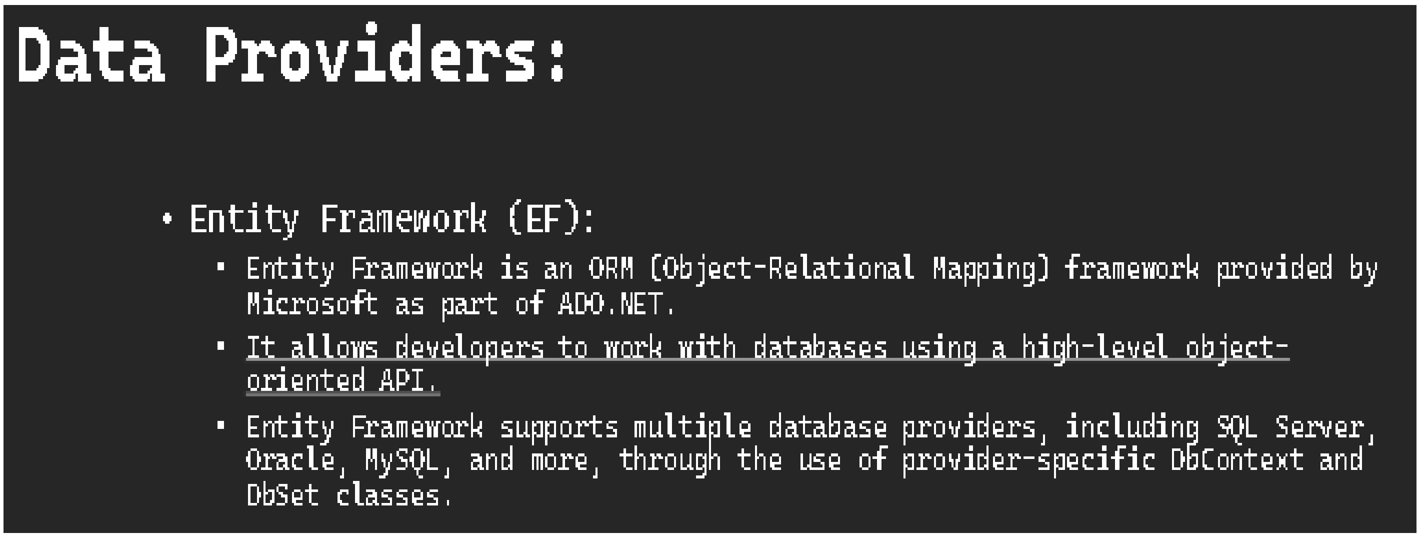
Data Providers:

- System.Data.SqlClient (SQL Server):
  - This data provider is specifically designed for Microsoft SQL Server databases.
  - It provides classes like SqlConnection, SqlCommand, SqlDataAdapter, and SqlDataReader to establish connections, execute commands, and retrieve data from SQL Server databases.
- System.Data.OracleClient (Oracle):
  - This data provider allows connectivity to Oracle databases.
  - It provides classes such as OracleConnection, OracleCommand, OracleDataAdapter, and OracleDataReader to work with Oracle databases.
- System.Data.MySql(MySql):
  - This data provider allows connectivity to MySql databases.



Data Providers:

- System.Data.OleDb (OLE DB):
  - OleDB (Object Linking and Embedding Database)
  - This data provider allows access to various databases through the OLE DB technology.
  - OLEDB is a Microsoft technology and is primarily used on Windows platforms.
  - It supports a wide range of databases, including Microsoft Access, Oracle, MySQL, and more.
  - It provides classes like OleDbConnection, OleDbCommand, OleDbDataAdapter, and OleDbDataReader for interacting with OLE DB data sources.
- System.Data.Odbc (ODBC):
  - ODBC (Open Database Connectivity) .
  - This data provider enables connectivity through the ODBC (Open Database Connectivity) standard.
  - ODBC is a widely adopted standard for accessing databases and is supported on various platforms, including Windows, macOS, and Linux.
  - It supports databases that comply with the ODBC standard, such as Microsoft SQL Server, Oracle, MySQL, and others.
  - It provides classes like OdbcConnection, OdbcCommand, OdbcDataAdapter, and OdbcDataReader for working with ODBC data sources.



Data Providers:

- Entity Framework (EF):
  - Entity Framework is an ORM (Object-Relational Mapping) framework provided by Microsoft as part of ADO.NET.
  - It allows developers to work with databases using a high-level object-oriented API.
  - Entity Framework supports multiple database providers, including SQL Server, Oracle, MySQL, and more, through the use of provider-specific DbContext and DbSet classes.

ADO.NET (ActiveX Data Objects for .NET) is a data access technology provided by Microsoft as part of the .NET Framework. It includes a set of data providers that enable developers to connect to and interact with different types of databases and data sources. Here are some commonly used data providers in ADO.NET:

Data provider is used to connect to the database, execute commands and retrieve the record. It is lightweight component with better performance. It also allows us to place the data into DataSet to use it further in our application.

The .NET Framework provides the following data providers that we can use in our application.

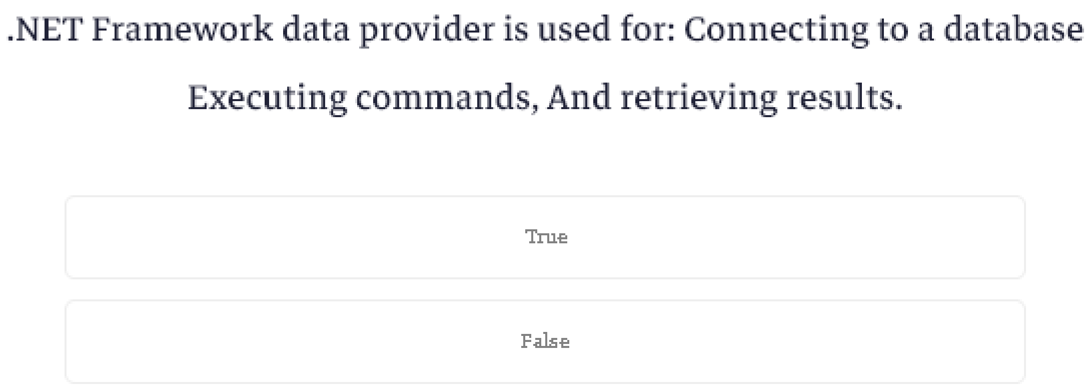
- **System.Data.SqlClient** (SQL Server):
  - This data provider is specifically designed for Microsoft SQL Server databases.
  - It provides classes like SqlConnection, SqlCommand, SqlDataAdapter, and SqlDataReader to establish connections, execute commands, and retrieve data from SQL Server databases.

- **System.Data.OracleClient** (Oracle):
  - This data provider allows connectivity to Oracle databases.
  - It provides classes such as OracleConnection, OracleCommand, OracleDataAdapter, and OracleDataReader to work with Oracle databases.
- **System.Data.OleDb** (OLE DB):
  - OleDB (Object Linking and Embedding Database)
  - This data provider allows access to various databases through the OLE DB technology.
  - OLEDB is a Microsoft technology and is primarily used on Windows platforms.
  - It supports a wide range of databases, including Microsoft Access, Oracle, MySQL, and more.
  - It provides classes like OleDbConnection, OleDbCommand, OleDbDataAdapter, and OleDbDataReader for interacting with OLE DB data sources.
- **System.Data.Odbc** (ODBC):
  - ODBC (Open Database Connectivity) .
  - This data provider enables connectivity through the ODBC (Open Database Connectivity) standard.
  - ODBC is a widely adopted standard for accessing databases and is supported on various platforms, including Windows, macOS, and Linux.
  - It supports databases that comply with the ODBC standard, such as Microsoft SQL Server, Oracle, MySQL, and others.
  - It provides classes like OdbcConnection, OdbcCommand, OdbcDataAdapter, and OdbcDataReader for working with ODBC data sources.
- **System.Data.MySql**(MySql):
  - This data provider allows connectivity to MySql databases.
- **Entity Framework (EF)**:
  - Entity Framework is an ORM (Object-Relational Mapping) framework provided by Microsoft as part of ADO.NET.
  - It allows developers to work with databases using a high-level object-oriented API.
  - Entity Framework supports multiple database providers, including SQL Server, Oracle, MySQL, and more, through the use of provider-specific DbContext and DbSet classes.

These are some of the data providers available in ADO.NET. The choice of data provider depends on the specific database system you are working with and the requirements of your application. ADO.NET provides a consistent programming model for accessing and manipulating data from various data sources.

ADO.NET Framework Data Providers.

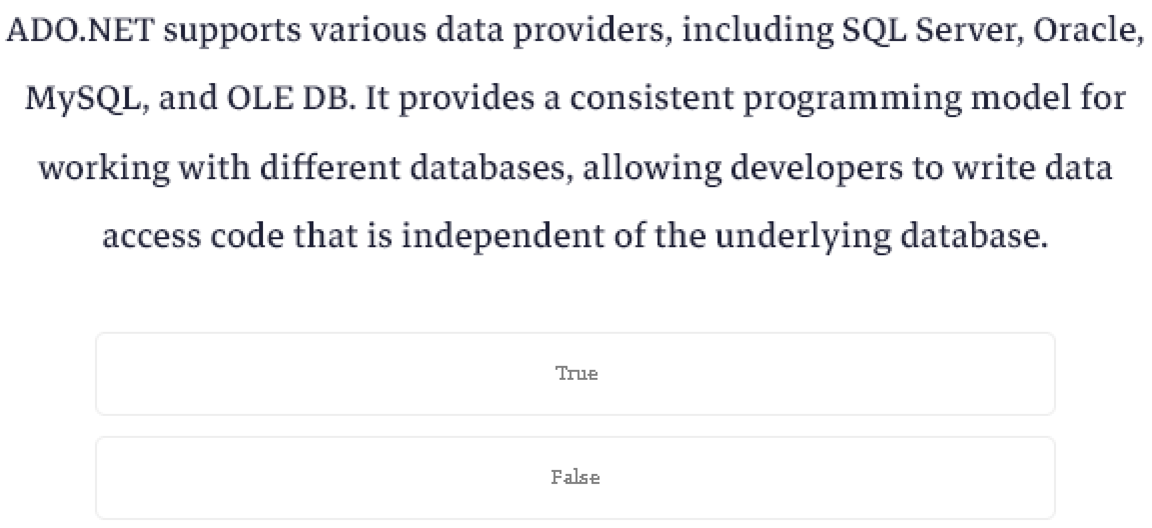
الواجب



.NET Framework data provider is used for: Connecting to a database Executing commands, And retrieving results.

True

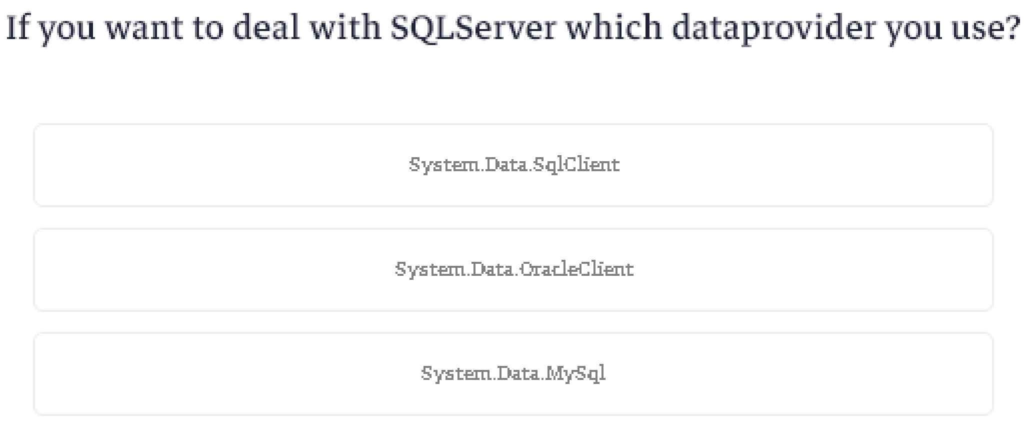
False



ADO.NET supports various data providers, including SQL Server, Oracle, MySQL, and OLE DB. It provides a consistent programming model for working with different databases, allowing developers to write data access code that is independent of the underlying database.

True

False

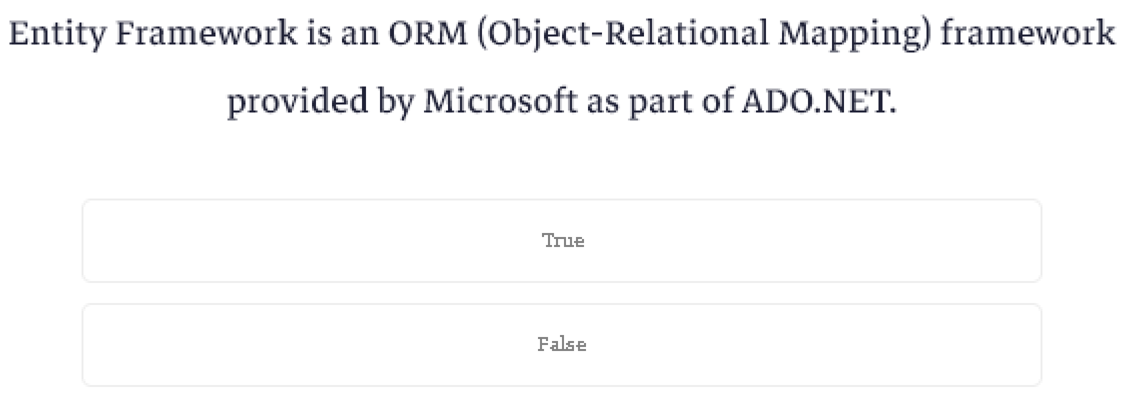


If you want to deal with SQLServer which dataprovider you use?

System.Data.SqlClient

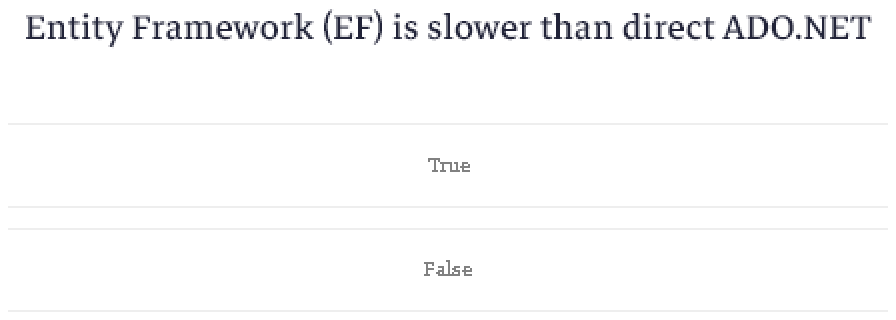
System.Data.OracleClient

System.Data.MySql


Entity Framework is an ORM (Object-Relational Mapping) framework provided by Microsoft as part of ADO.NET.

True

False


Entity Framework (EF) is slower than direct ADO.NET

True

False

ADO.NET Architecture (Components)

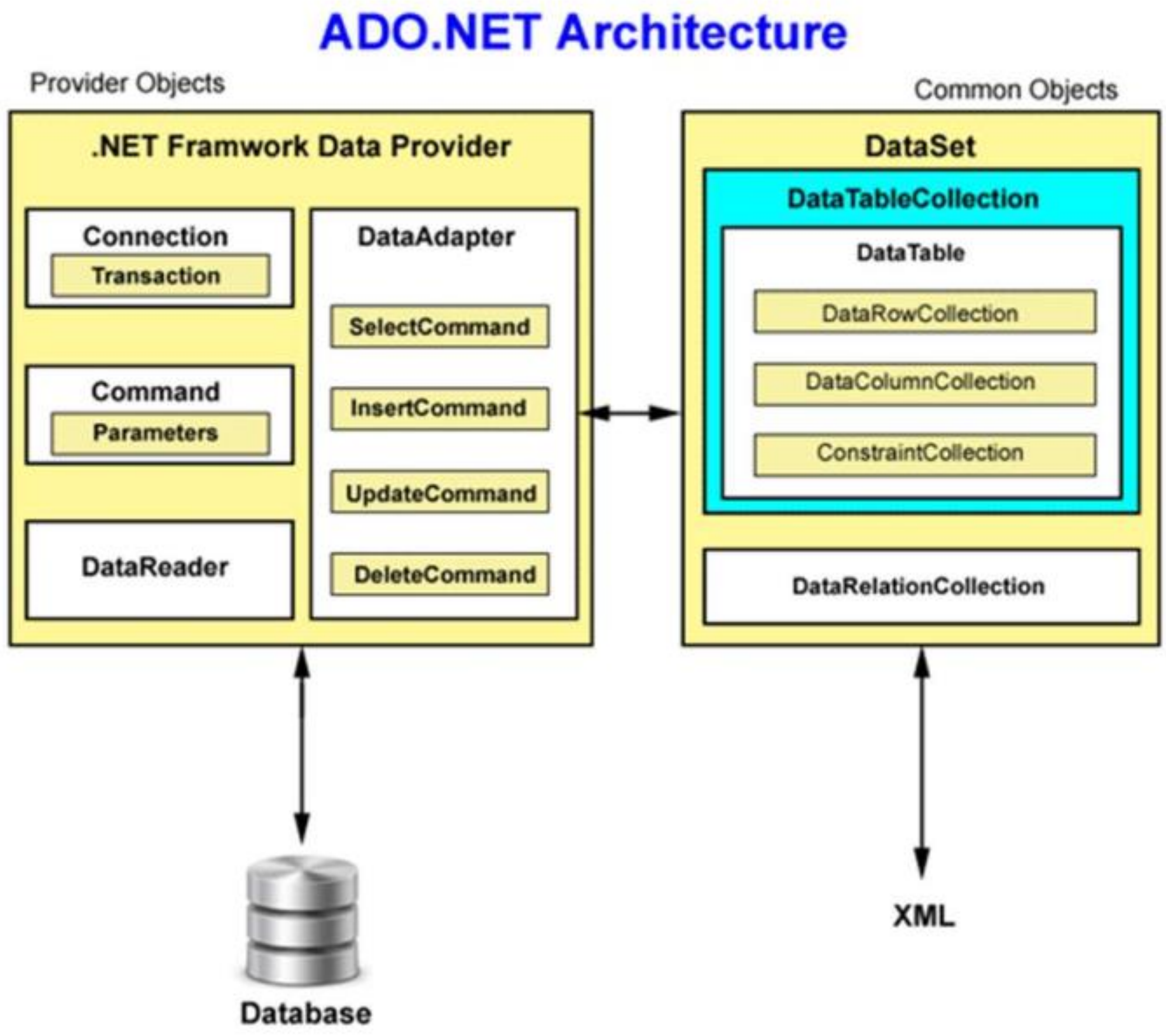
ال architecture بتاعت ال ado.net بتتكون من جزئين

اول جزء وهوا ال data provider وتاني جزء هوا ال dataset

ال data provider بينقسم لاربع أجزاء :-

1- Connection :- ودي اللي بتخلي البرنامج يوصل للداتا بيز وفيه بتديله البيانات بتاعت الداتا بيز اللي عايز تتواصل معاه وبياخد البيانات دي ويعمل establish connection
2- Command :- وده عن طريق بتقدر تبعت الأوامر من خلالها
3- Data reader :- ودي اغلب شغلك بيكون عن طريقها ودي بتكون read only و forward only يعني لما بتمشي عالداتا ب loop ماتقدرش ترجع تاني وهيا اسرع حاجه تخليك تقرا الداتا بيروح يجيبلك الداتا من غير مايحط عليها reference لانك مش هتقدر تعدل عليها
4- من مميزاتها كمان انها بترجعلك الداتا وبتفضل متوصله بالداتا بيز بعكس ال dataset
5- Data adapter :- وده عباره عن كوبري بين ال data reader وال data set ووظيفته انه ياخد الداتا اللي رجعت عن طريق ال data reader ويعبيها في الdata set وبعد ما ال data set تخلص شغلها بيرجع يعدل الداتا بيز

ال data set وهوا ابطأ من ال data reader لانه بيرجع الداتا كلها وبعدين يفصل الاتصال مع قاعدة البيانات يخليك تعدل عالداتا وتعمل فيها اللي انت عايزه وبعد ماتخلص يرجع يعمل اتصال مع قاعدة البيانات ويديها الداتا بعد التعديل ولا يفضل التعامل معها الا في حالات معينه



The main components of ADO.NET are:

- Connection: Represents a connection to a data source, such as a database. It provides methods to establish a connection, manage transactions, and execute commands.
- Command: Represents a query or a stored procedure that is executed against a data source. It provides methods to execute the command and retrieve the results.
- DataReader: Provides a fast, forward-only, read-only stream of data from a data source. It is used for retrieving large amounts of data efficiently and quickly.
- DataAdapter: Acts as a bridge between a dataset and a data source. It populates a dataset with data from the data source and also updates the data source with changes made to the dataset.
- DataSet: Represents an in-memory cache of data that can store multiple tables, relationships between tables, and constraints. It provides a disconnected representation of the data retrieved from a data source.

Components of ADO.NET

Components are designed for data manipulation and faster data access. **Connection, Command, DataReader, DataAdapter, DataSet,** and **DataView** are the components of ADO.NET that are used to perform database operations. ADO.NET has two main components that are used for accessing and manipulating data. They are as follows:

1. **Data Provider and**
2. **DataSet.**

.NET Framework Data Providers Objects

Following are the core object of Data Providers.

- **Connection**: It is used to establish a connection to a specific data source.
- **Command**: It is used to execute queries to perform database operations.
- **DataReader**: It is used to read data from data source. The DbDataReader is a base class for all DataReader objects.
- **DataAdapter**: It populates a **DataSet** and resolves updates with the data source. The base class for all DataAdapter objects is the DbDataAdapter class.

.NET Framework Data Provider for SQL Server

Data provider for SQL Server is a lightweight component. It provides better performance because it directly access SQL Server without any middle connectivity layer. In early versions, it interacts with ODBC layer before connecting to the SQL Server that created performance issues.

The .NET Framework Data Provider for SQL Server classes is located in the **System.Data.SqlClient** namespace. We can include this namespace in our C# application by using the following syntax.

```
using System.Data.SqlClient;
```

This namespace contains the following important classes.

- SqlConnection: It is used to create SQL Server connection. This class cannot be inherited.
- SqlCommand: It is used to execute database queries. This class cannot be inherited.
- SqlDataAdapter: It represents a set of data commands and a database connection that are used to fill the DataSet. This class cannot be inherited.
- SqlDataReader: It is used to read rows from a SQL Server database. This class cannot be inherited.
- SqlException: This class is used to throw SQL exceptions. It throws an exception when an error is occurred. This class cannot be inherited.

الواجب

Connection: Represents a connection to a data source, such as a database. It provides methods to establish a connection, manage transactions, and execute commands.

True

False

Command: Represents a query or a stored procedure that is executed against a data source. It provides methods to execute the command and retrieve the results.

True

False

DataReader: Provides a fast, forward-only, read-only stream of data from a data source. It is used for retrieving large amounts of data efficiently and quickly.

True

False

DataAdapter: Acts as a bridge between a dataset and a data source. It populates a dataset with data from the data source and also updates the data source with changes made to the dataset.

True

False

DataSet: Represents an in-memory cache of data that can store multiple tables, relationships between tables, and constraints. It provides a disconnected representation of the data retrieved from a data source.

True

False

Which is faster Data Reader or DataSet?

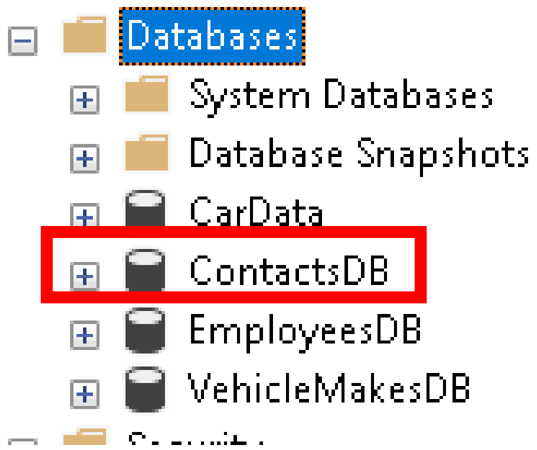
DataReader

DataSet

ADO.NET Code: Get ALL Contacts.

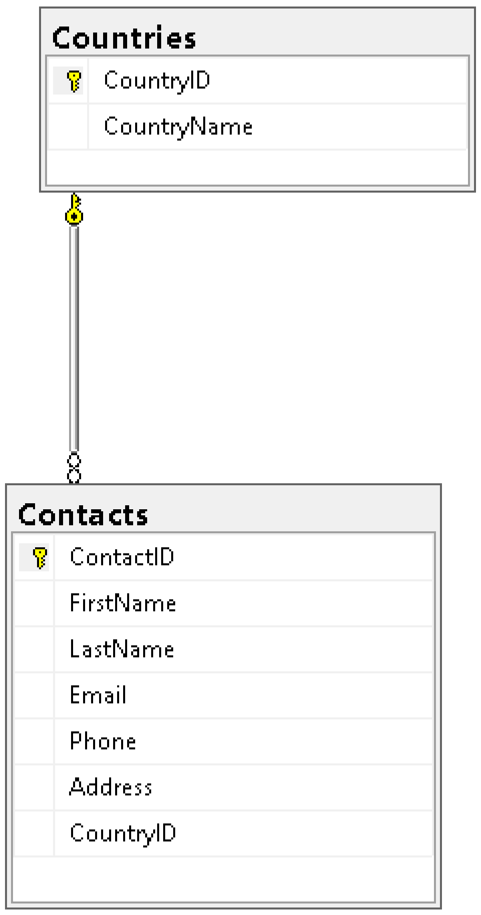
هنشتغل علي داتا بيز فيها ال contacts او دليل الهاتف بالعربي و هنبدأ نعمل عليها عمليات

اول حاجه هنعمل restore للداتا بيز علي ال sql server



```sql
use ContactsDB;
EXEC sp_changedbowner 'sa';
```

هنا الداتا بيز مربوطه مع جدول فيه اسامي الدول



هنبدأ نقرأ الداتا اللي في الجداول دي في ال c# في ال console application عشان نفهم الأول بنربط ازاي الداتا بيز وبعدين نشتغل

طيب دلوقتي الداتا بيز ممكن تكون عالجهاز عندي او علي سيرفر معين فعشان اعمل connection اول حاجه محتاجها هيا connection string وده بيكون فيه كل المعلومات اللي محتاجها عشان أوصل للداتا بيز

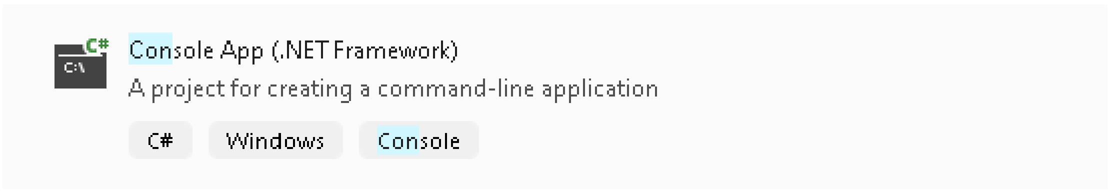
طيب كلنا عارفين اننا عشان نتواصل مع الداتا بيز محتاجين نستخدم ال ado.net ونحدد ال provider بتاعنا

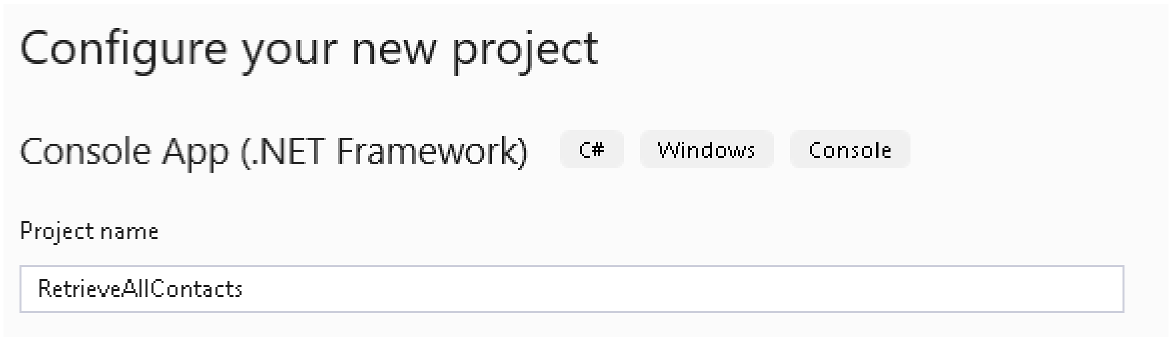
اول حاجه عاوزين نحددها هيا نوع قاعدة البيانات المستخدمه عشان نحدد ال provider اللي هنستعمله

في الحاله بتاعتنا الداتا بيز نوعها sql server يبقي هنستخدم ال sqlClient

ازاي ؟

تعالي الأول نعمل console application


Console App (.NET Framework)
A project for creating a command-line application


Configure your new project

Console App (.NET Framework)

Project name

RetrieveAllContacts

اول حاجه هيا انك تعمل implement للمكتبات بتاعت ال sql client

```
using System.Data.SqlClient;
```

تاني حاجه هنعرف ال connection string بره ال main وجوه كلاس الprogram و هيكون عباره عن static string لانه هيكون ثابت في المشروع وده اللي هنخزن فيه معلومات الداتابيز اللي هنتواصل معاها

طيب ازاي هنكتبها؟

قالك انه ال connection string بيتكون من اربع أجزاء كل بيفصل بينهم فاصله منقوطه اللي هيا دي (;) الأجزاء دي هيا كالاتي :-

1- ال server :- وده عنوان السيرفر اللي موجود عليه الداتا بيز ولو الداتا بيز عالجهاز بتاعك يعني local host بتكتب نقطه ولو علي سيرفر خارجي هتكتب ال ip address بتاع السيرفر ده
2- Database :- وده اسم الداتابيز نفسها
3- User & password :- ودول هما اليوزر والباسورد اللي عملناهم واحنا بنثبت ال sql server ومن غيرهم مش هتقدر تدخل عالداتا بيز

اليوزر بتاعنا هوا sa والباسورد sa123456

بس هتحط كل الداتا دي في ال string اللي عملناه زي كده

```
static string ConnectionString = "Server=.;Database=ContactsDB;User id=sa;Password=sa123456";
```

طيب احنا دلوقتي عاوزين نعمل function تطبعلنا كل البيانات اللي موجوده في جدول ال contacts

عشان نعمل ده محتاجين نتصل بالداتا بيز ونبعتلها ال query اللي عاوزين ننفذها وبعدين نقرأ النتايج

اول حاجه عاوزين نربط التطبيق بتاعنا بالداتا بيز الكلاس المسؤول عن ال connection ده اسمه SqlConnection بناخد منه object ونديله ال connection string اللي عملناه

```
SqlConnection Connection = new SqlConnection(ConnectionString);
```

تاني حاجه هيا امر ال sql ا ال query اللي عاوزين ننفذها ودي هتكون عباره عن متغير من النوع string هنحط فيه ال query بتاعتنا

```
string Query = "select * from Contacts";
```

تالت حاجه اننا ننفذ ال query فهناخد object من كلاس اسمه SqlCommand ونديله ال query اللي عملناها و ال connection اللي عملناه

```
SqlCommand Command = new SqlCommand(Query, Connection);
```

لحد كده احنا نفذنا امر ال sql

دلوقتي بقي عاوزين نقرأ الداتا اللي رجعت

فقالك قبل ماتعمل أي حاجه لازم تعمل try& catch عشان لو حصل exception ولا حاجه يطلعلك الرساله بتاعت الخطأ

ال try & catch هنا اللي اعرفه انها بتتعمل في العاده كل ماتيجي تفتح connection مع داتابيز او file

```
try {

}catch(Exception ex) {
Console.WriteLine(ex.Message);
}
```

طيب اول حاجه عاوزين نعملها جوه ال try هيا انك تفتح الاتصال مع الداتابيز

طب احنا مش فتحناه قبل كده؟

لا احنا اللي عملناه قبل كده اننا كنا بنأسس للاتصال

طيب نعمل ايه عشان نتصل بالداتابيز

هنستدعي function اسمها open من ال object اللي اسمه connection

```
Connection.Open();
```

طيب لحد دلوقتي احنا نفذنا ال query عاوزين قرأ الداتا بقي هنقراها ازاي؟

فاكر لما كنا في مشروع البنك وكنا بنقرأ الداتا كنا بنعمل while loop ونقوله getline فيقوم يقرأ في كل مره سطر ناخده نخزنه في ال vector

هنا نفس الكلام

بس بدل مانستخدم get line لا فيه حاجه شبها في ال architecture components اسمها data reader وقولنا ان دي المسؤوله عن انها تاخد الداتا من قاعدة البيانات وتعرضهالك في شكل

read only وبتكون زي المؤشر كده بيقف علي كل record في الجدول ويرجعهولك وقولنا انه ده الأسرع في التعامل مع الداتا بيز

هنستخدمه ازاي؟

اول حاجه هنعمل object من كلاس اسمه SqlDataReader بس مش هنستدعي ال CONSTRUCTOR بتاعه هنستعدي method موجوده في ال object اللي اخدناه من ال sqlCommand اسمها ExecuteReader

```
SqlDataReader Reader = Command.ExecuteReader();
```

بعدين هنعمل while loop ونقوله reader.read

```
while (Reader.Read())
{
}
```

ال reader هنا بيرجع array كل العناصر اللي فيه بتعبر عن قيمة عمود معين في الجدول فالنت في الخطوه دي كل اللي بتعمله انك بتعين متغير لكل عمود وبتخزن فيه القيمه اللي جايالك من ال reader

بتستدعي القيمه عن طريق انك تكتب رقم العمود او انك تكتب اسمه في الداتا بيز بس يفضل انك تكتب اسمه عشان اللخبطه

هوا هنا بدل مايخزن الداتا كلها في array قام جاي طابعها عالشاشه اول بأول للتبسيط

وطبعا طالما فتحت connection لازم تقفله وبتقفل ال reader

ده الجزء الخاص بال while loop

```
try {
```

```
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

while (Reader.Read())
{
int ContactID = (int)Reader["ContactID"];
string FirstName=(string)Reader["FirstName"];
string LastName=(string)Reader["LastName"];
string Email=(string)Reader["Email"];
string Phone=(string)Reader["Phone"];
string Address=(string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();

}
Reader.Close();
Connection.Close();
}
```

وده الكود كله

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";

static void PrintAllContacts()
{

SqlConnection Connection = new SqlConnection(ConnectionString);
string Query = "select * from Contacts";
SqlCommand Command = new SqlCommand(Query, Connection);


try
{
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();
while (Reader.Read())
{
int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();

}
Reader.Close();
Connection.Close();
}
catch (Exception ex)
{
Console.WriteLine(ex.Message);
}

}
static void Main(string[] args)
{
PrintAllContacts();
Console.ReadLine();
}
}
```

Parameterized Query

عاوزين نضيف شروط عال query بتاعتنا قالك عندك طريقتين الاولي انك تعمل concatenation بين متغيرين من النوع string (يعني تكتب ال query علي بعضها) ياما تستخدم ال parameterized query

قبل ماتعمل حاجه هنضيف اتنين records عالجدول بتاع ال contacts

```
insert into Contacts
VALUES('Jane','Brown','davidbrown@email.com','999888777','654Cedar St',1),
('Jane','Doe','jj@jj.com','1123413','1234',1)
```

طيب تعالي بقي للكود

عاوزين نعدل في الطباعه بحيث انه ال method نفسها تاخد fisrt name وبعدين نضيف شرط لل query انه ال first name يكون هوا نفسه ال first name اللي جاي من ال method

عشان نعمل كده عندنا طريقتين

اول طريقه اننا نكتب ال query علي بعضها زي كده

```
static void PrintAllContacts(string FirstName1)
```

```
{

SqlConnection Connection = new SqlConnection(ConnectionString);
string Query = "select * from Contacts where FirstName="+FirstName1;
SqlCommand Command = new SqlCommand(Query, Connection);
```

تاني طريقه ودي اللي يفضل انك تستخدمها وهيا عن طريق ال parameterized query

ودي بتتم عن طريق خطوتين

الخطوه الاولي انك تكتب الشرط عادي في ال query بس بدل ماتحط اسم المتغير اللي جاي من ال method بره ال string لا هتحطه جوه ال string وقبليه علامة @ زي كده

```
string Query = "select * from Contacts where FirstName=@FirstName1";
```

تاني خطوه وهيا انك تستدعي method اسمها add with value وبتديها string بيكون عباره عن اسم المتغير اللي قبله @ واسم المتغير اللي جاي من ال method كأنك بتستبدل جزء من الtext بقيمة المتغير يعني

```
Command.Parameters.AddWithValue("@FirstName1", FirstName1);
```

ده الكود كله

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";

static void PrintAllContacts(string FirstName1)
{

SqlConnection Connection = new SqlConnection(ConnectionString);
string Query = "select * from Contacts where FirstName=@FirstName1";
SqlCommand Command = new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue("@FirstName1", FirstName1);


try
{
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

while (Reader.Read())
{
int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();

}
Reader.Close();
Connection.Close();
}
catch (Exception ex)
{
Console.WriteLine(ex.Message);
}

}
static void Main(string[] args)
{
PrintAllContacts("Jane");
Console.ReadLine();
}
}
```

ولو عايز اضيف كمان parameter بكتب and في الquery وبستدعي ال add with value تاني

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";

static void PrintAllContacts(string FirstName1,int CountryID1)
{

SqlConnection Connection = new SqlConnection(ConnectionString);
string Query = "select * from Contacts where FirstName=@FirstName1 and CountryID=@CountryID1";
SqlCommand Command = new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue("@FirstName1", FirstName1);
Command.Parameters.AddWithValue("@CountryID1", @CountryID1);


try
{
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

while (Reader.Read())
{
int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();
```

```
            }
            Reader.Close();
            Connection.Close();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }

    }
    static void Main(string[] args)
    {
        PrintAllContacts("Jane",1);
        Console.ReadLine();
    }
}
```

Parameterized Query With "Like"

عشان تعمل query فيها like افتكر اننا كنا لما بنكتب أي نص في ال sql server كنا بنكتبه بين single quotes مش double

طريقته هيا نفسها طريقة ال parameterized query بس في جملة ال sql بتيجي قبل وبعد ال parameter وتحط single quotes وعندك الحريه انك تسيبهم فاضيين او تحط فيهم العلامه %

لو بتدور في بداية الكلمه

```
string Query = "Select * from Contacts Where FirstName like " + @StartsWith + '%' ";

SqlCommand Command = new SqlCommand(Query,Connection);

Command.Parameters.AddWithValue("@StartsWith", StartsWith);
```

لو بتدور في نهاية الكلمة

```
string Query = "select * from Contacts where FirstName like '%' + @EndsWith + "";

SqlCommand command = new SqlCommand(Query,connection);
command.Parameters.AddWithValue("@EndsWith", EndsWith);
```

لو بتدور في نص الكلمه

```
string Query = "select * from Contacts where FirstName like '%' + @Contains + '%'";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@Contains", Contains);
```

لو بتدور باكتر من مقطع زي مثلا بتدور علي احمد محمد إبراهيم وعايز تكتب في البحث (حم مح يم) بتستخدم ال contains عادي بس بتيجي وانت بتستدعي ال function تكتب العلامه % مكان المقطع اللي ناقص من الكلمه زي كده (حم%مح%يم) بس ده اليوزر اللي هيستخدم البحث هوا اللي هيعمل كده او انت ممكن تعمله بانك تعمل بحث ب 3 مقاطع وتحط 3 متغيرات بس الاحسن اليوزر هوا اللي يعملها

وده الكود كله

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";


static void SearchContactsStartsWith(string StartsWith) {
SqlConnection Connection = new SqlConnection(ConnectionString);
string Query = "Select * from Contacts Where FirstName like " + @StartsWith + '%' ";

SqlCommand Command = new SqlCommand(Query,Connection);

Command.Parameters.AddWithValue("@StartsWith", StartsWith);

try {
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

while (Reader.Read()) {

int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();
}

Reader.Close() ;
Connection.Close();
} catch (Exception ex) {
Console.WriteLine(ex.Message);
}
}

static void SearchContactsEndsWith(string EndsWith)
{
SqlConnection connection=new SqlConnection(ConnectionString);
string Query = "select * from Contacts where FirstName like '%' + @EndsWith + "";

SqlCommand command = new SqlCommand(Query,connection);
command.Parameters.AddWithValue("@EndsWith", EndsWith);

try {
connection.Open();
SqlDataReader Reader = command.ExecuteReader();

while (Reader.Read()) {
int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();
}

Reader.Close();
connection.Close();
}catch (Exception ex) {
Console.WriteLine(ex.Message); }

}

static void SearchContactsContains(string Contains) {
SqlConnection connection = new SqlConnection(ConnectionString);
string Query = "select * from Contacts where FirstName like '%' + @Contains + '%'";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@Contains", Contains);

try
{
connection.Open();
SqlDataReader Reader = command.ExecuteReader();

while (Reader.Read())
{
int ContactID = (int)Reader["ContactID"];
string FirstName = (string)Reader["FirstName"];
string LastName = (string)Reader["LastName"];
string Email = (string)Reader["Email"];
string Phone = (string)Reader["Phone"];
string Address = (string)Reader["Address"];
int CountryID = (int)Reader["CountryID"];

Console.WriteLine($"Contact ID: {ContactID}");
Console.WriteLine($"First Name: {FirstName}");
Console.WriteLine($"Last Name: {LastName}");
Console.WriteLine($"Email: {Email}");
Console.WriteLine($"Phone: {Phone}");
Console.WriteLine($"Address: {Address}");
Console.WriteLine($"Country ID: {CountryID}");
Console.WriteLine();
}

Reader.Close();
connection.Close();
}
catch (Exception ex)
{
Console.WriteLine(ex.Message);
}
}

static void Main(string[] args)
{
Console.WriteLine("--------Contacts Starts With 'j'");
SearchContactsStartsWith("j");

Console.WriteLine("--------Contacts Ends With 'ne'");
SearchContactsEndsWith("ne");
```

```
Console.WriteLine("--------Contacts Contains 'ae'");
SearchContactsContains("ae");
Console.ReadLine();
}
}
```

Retrieve a Single Value (ExecuteScalar)

ال excute reader بترجعلك اكتر من record

لكن لو ال query بتاعتك بترجعلك قيمه واحده بس تقدر تستخدم حاجه اسمها execute scalar ودي بترجعلك قيمة اول خليه في الجدول اللي بتكون موجوده في اول صف وأول عمود

طريقتها نفس الطريقه بتاعت ال reader بتختلف ففي السطر بتاع ال execute reader

بتاخد object من كلاس اسمه object وتستدعي ال function اللي اسمها executeScalar وبعدين بتعمل if statement عالقيمه اللي طالعه عشان لو كانت القيمه ب null مايعملش exception وماتجيش جواها وتعمل return لانك كده مش هتوصل للسطر اللي بيقفل الاتصال فهيفضل مفتوح

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";


static string GetFirstName(int ContactID)
{
string FirstName="";
SqlConnection connection = new SqlConnection(ConnectionString);
string Query = "select FirstName from Contacts where ContactID = @ContactID";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@ContactID", ContactID);

try
{
connection.Open();
object result= command.ExecuteScalar();
if (result != null ) { FirstName= result.ToString(); }else { FirstName= ""; }

connection.Close();
}
catch (Exception ex) { Console.WriteLine(ex.Message); }

return FirstName;
}
static void Main(string[] args)
{
Console.WriteLine(GetFirstName(1));

Console.ReadLine();
}
}
```

Find Single Contact

هنا احنا عاوزين نعمل structure لل contacts عشان نقدر ناخد منه متغيرات ونقدر نخزن الداتا في البرنامج بتاعنا ونقدر نتحكم فيها اكتر وهنعمل function تاخد ال id وترجع البيانات بتاعت contact كلها زي ماكنا بنعمل في مشروع البنك بس دلوقتي بالداتا بيز

فاول حاجه هنعمل ال structure

```
public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}
```

والباقي هنشتغل عليه زي ما كنا بنعمل في مشاريع ال c++ هيختلف بس حته اننا بدل ماكنا بنتواصل مع ملف نصي لا هنتواصل مع داتا بيز

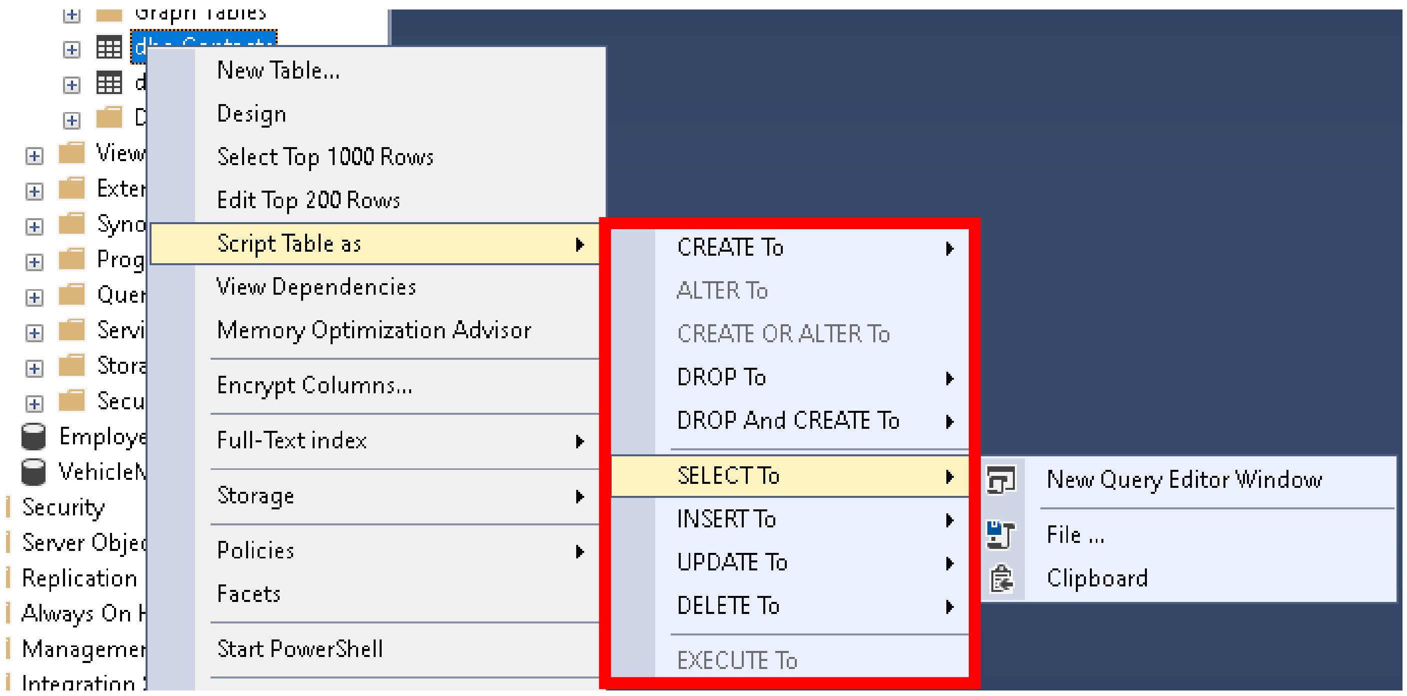
هنا مش هنحتاج while loop عال reader لانه بيرجع record واحد بس

```
using System;

using System.Data.SqlClient;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";

static bool FindContactByID(int ContactID,ref stContact ContactInfo) {
bool isFound = false;

SqlConnection connection=new SqlConnection(ConnectionString);
string Query = "select * from Contacts where ContactID = @ContactID";
SqlCommand command = new SqlCommand(Query,connection);
command.Parameters.AddWithValue("@ContactID", ContactID);

try {
connection.Open();
SqlDataReader reader = command.ExecuteReader();

if (reader.Read())
{
isFound = true;

ContactInfo.ID = (int)reader["ContactID"];
ContactInfo.FirstName = (string)reader["FirstName"];
ContactInfo.LastName = (string)reader["LastName"];
ContactInfo.Email = (string)reader["Email"];
ContactInfo.Phone = (string)reader["Phone"];
ContactInfo.Address = (string)reader["Address"];
ContactInfo.CountryID = (int)reader["CountryID"];

}
else { isFound = false; }
reader.Close();
connection.Close();
}catch(Exception ex) { Console.WriteLine(ex.Message); }

return isFound;
}

public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}
static void Main(string[] args)
{
stContact ContactInfo= new stContact();

if (FindContactByID(1,ref ContactInfo)) {
Console.WriteLine($"\nContact ID: {ContactInfo.ID}");
Console.WriteLine($"Name: {ContactInfo.FirstName}");
Console.WriteLine($"Email: {ContactInfo.Email}");
Console.WriteLine($"Phone: {ContactInfo.Phone}");
Console.WriteLine($"Address: {ContactInfo.Address}");
Console.WriteLine($"CountryID: {ContactInfo.CountryID}");

} else {
Console.WriteLine("Contact not Found");
}

Console.ReadLine();
}
}
```

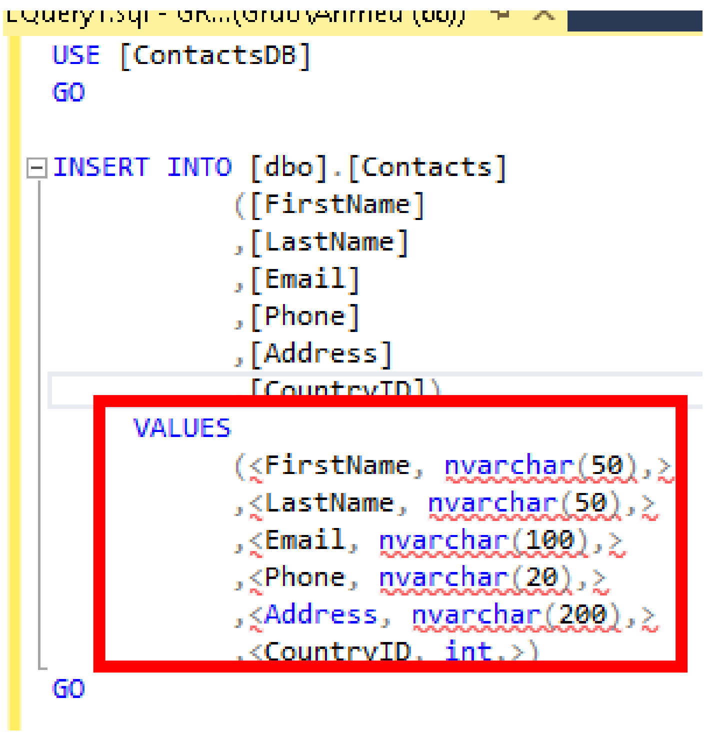
Insert-Add Data

قبل مانبدأ هوا هنا بيديك اختصار بدل ماتروح في ال sql server وتكتب أوامر الاضافه والتعديل ممكن تعمل حركه بالماوس تجيبلك الكود جاهز وتروح انت تحط القيم بايدك وخلاص

بتعمل كده عن طريق انك تروح لجدول اللي عاوز تنفذ علي الامر بتاعك وتدوس كليك يمين



بتختار الامر اللي انت عاوزه وبعدين من القايمه اللي بعدها بتختار new query editor وبتلاقي الكود جاهز



```
USE [ContactsDB]
GO

INSERT INTO [dbo].[Contacts]
           ([FirstName]
           ,[LastName]
           ,[Email]
           ,[Phone]
           ,[Address]
           ,[CountryID])
     VALUES
           (<FirstName, nvarchar(50),>
           ,<LastName, nvarchar(50),>
           ,<Email, nvarchar(100),>
           ,<Phone, nvarchar(20),>
           ,<Address, nvarchar(200),>
           ,<CountryID, int,>)
GO
```

دلوقتي هوا بيقولك انه أوامر ال sql بتكون نوعين النوع الأول اللي بتستخدم فيه ال excute query وده الخاص باوامر ال select statement ودي اللي بتستخدمها عشان ترجع او تستعلم عن بيانات معينه في الداتابيز

النوع التاني وهوا ال excute none query ودي العمليات اللي بتعملها علي البيانات من حذف واضافه وتعديل

لو ماعديتش علي الكلام اللي انا كاتبه في المقدمه بتاعت الكورس روح بص عليها التعقيدات هتتبسط معاك

طيب احنا دلوقتي لما نحب نضيف داتا في جدول معين هنستخدم ال excute non query

تعالي نشوف ازاي :-

اول حاجه طبعا لازم نسهل علي نفسنا طريقة التعامل مع الداتا والتحكم فيها فهنعمل لل structure contacts

```
public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}
```

تعالي في ال main ناخد منه variable ونعبي فيه الداتا ونستدعي function تضيف الداتا

يعني نجهز الداتا اللي عاوزين نبعتها

```
static void Main(string[] args)
{
stContact contact = new stContact {
FirstName="Mohammed",
LastName="Abu-Hadhoud",
Email="m@example.com",
Phone="1234567890",
Address="123 Main Street",
CountryID=1,
};

AddNewContact(contact);

Console.ReadLine();
}
}
```

عملنا ال structure وجهزنا الداتا فاضل ايه؟

فاضل اننا نوصل للداتابيز ونضيف الداتا للجدول وده هيتم في ال function اللي اسمها add new contact

اول 3 سطور هما هما بتوع ال connection وال query وال command وهتعمل add with value بعدد المتغيرات اللي عايز تدخلها

```
static void AddNewContact(stContact newContact) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = "insert into Contacts (FirstName,LastName,Email,Phone,Address,CountryID)" +
"Values(@FirstName,@LastName,@Email,@Phone,@Address,@CountryID)";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@FirstName",newContact.FirstName);
command.Parameters.AddWithValue("@LastName", newContact.LastName);
command.Parameters.AddWithValue("@Email", newContact.Email);
command.Parameters.AddWithValue("@Phone", newContact.Phone);
command.Parameters.AddWithValue("@Address", newContact.Address);
command.Parameters.AddWithValue("@CountryID", newContact.CountryID);

try {
```

```
connection.Open();

} catch (Exception ex) { Console.WriteLine(ex.Message); }

}
```

بعدين ندخل جوه ال try

ال object اللي اخدناه بتاع ال connection زي ما كان فيه ال excute reader وال excute scalar برضه فيه function اسمها executeNonQuery ودي بتبعت ال query لل sql ينفذه وبترجعلك int بيمثل عدد ال records اللي اتأثرت بال query بتاعتك

طيب هتاخد ال query منين ؟

ماالحنا لما جينا اخدنا ال object من ال command ال constructor بتاعه اخد ال query

طيب الرقم اللي راجع من ال function دي لو كان بصفر وال query بتاعتنا أصلا كانت عباره عن اضافه record جديد يبقي العمليه ماتمتش ولو اكبر من صفر يبقي تم بنجا ممكن نخد الكلام ده ونحطه في if statement

```
int RowsAffected=command.ExecuteNonQuery();

if (RowsAffected > 0) { Console.WriteLine("Record inserted Successfully"); }
else { Console.WriteLine("Record insertion Faild!"); }
```

ده الكود كله

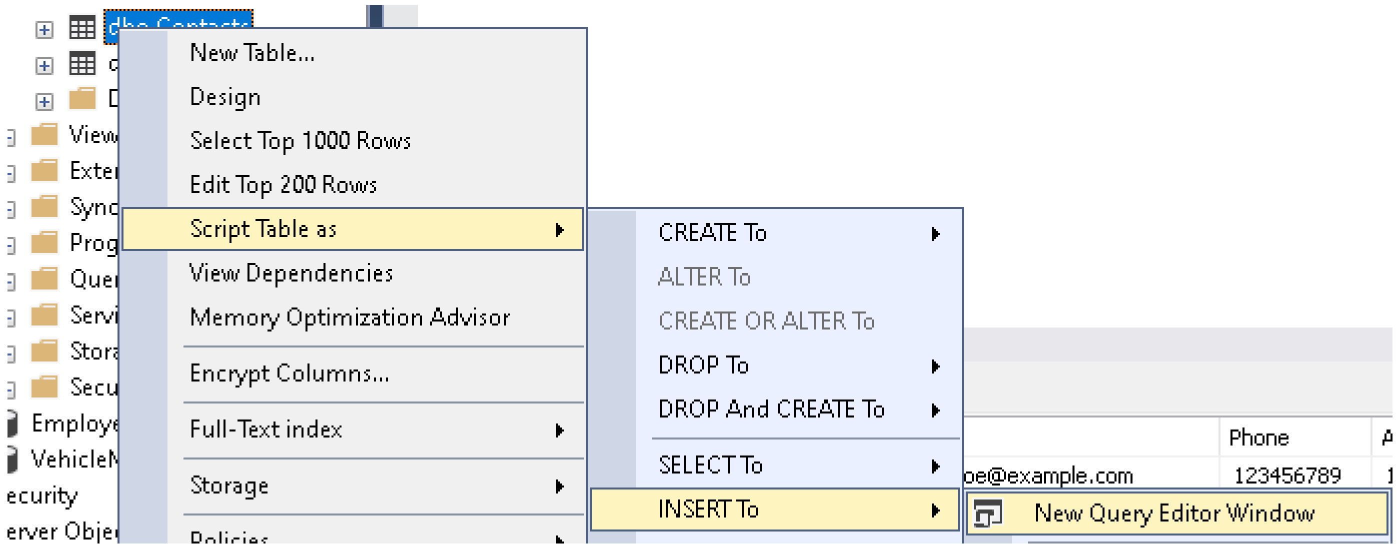
```
using System;

using System.Data.SqlClient;
using System.Runtime.CompilerServices;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";



public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}


static void AddNewContact(stContact newContact) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = "insert into Contacts (FirstName,LastName,Email,Phone,Address,CountryID)" +
"Values(@FirstName,@LastName,@Email,@Phone,@Address,@CountryID)";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@FirstName",newContact.FirstName);
command.Parameters.AddWithValue("@LastName", newContact.LastName);
command.Parameters.AddWithValue("@Email", newContact.Email);
command.Parameters.AddWithValue("@Phone", newContact.Phone);
command.Parameters.AddWithValue("@Address", newContact.Address);
command.Parameters.AddWithValue("@CountryID", newContact.CountryID);

try {

connection.Open();
int RowsAffected=command.ExecuteNonQuery();

if (RowsAffected > 0) { Console.WriteLine("Record inserted Successfully"); }
else { Console.WriteLine("Record insertion Faild!"); }
connection.Close();
} catch (Exception ex) { Console.WriteLine(ex.Message); }

}


static void Main(string[] args)
{
stContact contact = new stContact {
FirstName="Mohammed",
LastName="Abu-Hadhoud",
Email="m@example.com",
Phone="1234567890",
Address="123 Main Street",
CountryID=1,
};

AddNewContact(contact);

Console.ReadLine();
}
}
```

Retrieve Auto Number after Inserting/Adding Data

دلوقتي انا بعد ماضيفت ال record فيه عمود ال id بيكون auto generated انا بقي عايز ارجعه عشان استخدمه في حاجات تانيه

طيب انا دلوقتي هضيف record جديد باستخدام ال sql و هعرف ازاي هرجع ال id وبعدين هنطبق عال c#

يلا نجهز ال record اللي هنضيفه

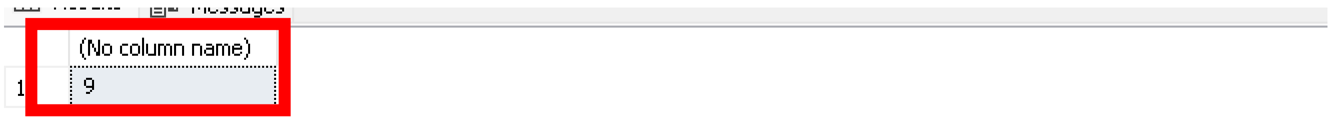


```
USE [ContactsDB]
GO

INSERT INTO [dbo].[Contacts]
([FirstName]
,[LastName]
,[Email]
,[Phone]
,[Address]
,[CountryID])
VALUES
('Ali','Omar','A@a','03993992','123 stree',1)
GO
```

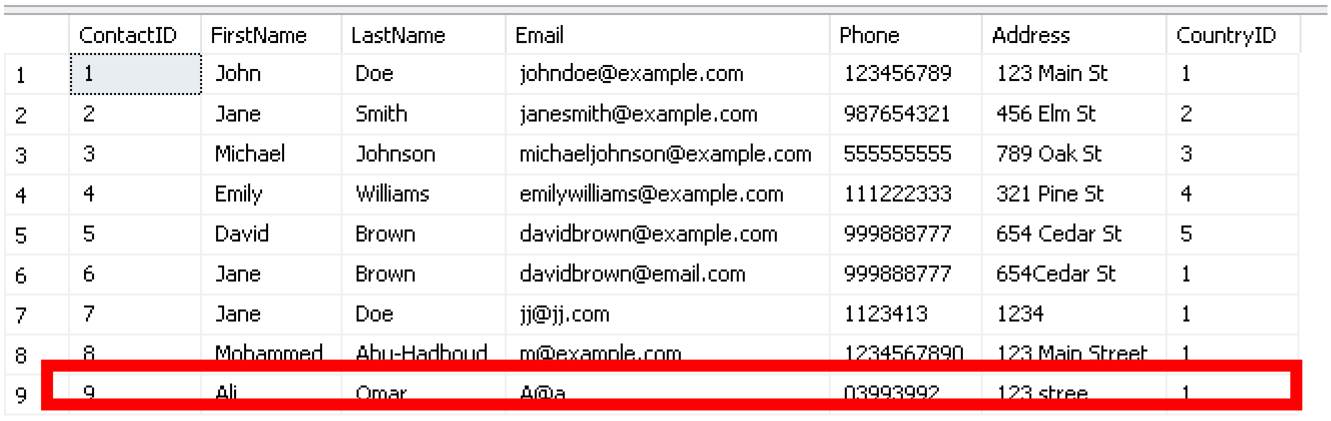
طيب عشان نرجع ال id فيه في ال sql server حاجه اسمها SCOPE_IDENTITY ودي function بترجع ال identity اللي هوا ال id يعني بس within scope يعني في اطار معين او تحت scope معين

فلو كتبت امر ال sql واستدعيت ال function دي وشغلت الاتنين مع بعض هينفذ امر ال sql ويرجعلك ال id

```
USE [ContactsDB]
GO

INSERT INTO [dbo].[Contacts]
([FirstName]
,[LastName]
,[Email]
,[Phone]
,[Address]
,[CountryID])
VALUES
('Ali','Omar','A@a','03993992','123 stree',1)
GO

select SCOPE_IDENTITY()

select* from Contacts
```



	(No column name)
1	9



	ContactID	FirstName	LastName	Email	Phone	Address	CountryID
1	1	John	Doe	johndoe@example.com	123456789	123 Main St	1
2	2	Jane	Smith	janesmith@example.com	987654321	456 Elm St	2
3	3	Michael	Johnson	michaeljohnson@example.com	555555555	789 Oak St	3
4	4	Emily	Williams	emilywilliams@example.com	111222333	321 Pine St	4
5	5	David	Brown	davidbrown@example.com	999888777	654 Cedar St	5
6	6	Jane	Brown	davidbrown@email.com	999888777	654Cedar St	1
7	7	Jane	Doe	jj@jj.com	1123413	1234	1
8	8	Mohammed	Abu-Hadhoud	m@example.com	1234567890	123 Main Street	1
9	9	Ali	Omar	A@a	03993992	123 stree	1

يلا نعمل الحوار ده في ال c#

احنا دلوقتي عندنا جملتين sql عاوزين ننفذهم مره واحده

فكل اللي هنعمله اننا هنفصل بالجملتين عن طريق ال semicolon او الفاصله المنقوطه

و هننفذ ال query باستخدام ال scalar لانها بتقدر تنفذ امر الsql مهما كان حجمه وبترجعلك اول عمود من اول صف

```
using System;

using System.Data.SqlClient;
using System.Runtime.CompilerServices;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";


public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}


static void AddNewContactAndGetID(stContact newContact) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = "insert into Contacts (FirstName,LastName,Email,Phone,Address,CountryID)" +
"Values(@FirstName,@LastName,@Email,@Phone,@Address,@CountryID) ; " +
"select SCOPE_IDENTITY()";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@FirstName",newContact.FirstName);
command.Parameters.AddWithValue("@LastName", newContact.LastName);
command.Parameters.AddWithValue("@Email", newContact.Email);
command.Parameters.AddWithValue("@Phone", newContact.Phone);
command.Parameters.AddWithValue("@Address", newContact.Address);
command.Parameters.AddWithValue("@CountryID", newContact.CountryID);

try {

connection.Open();
object result=command.ExecuteScalar();

if (result !=null && int.TryParse(result.ToString(),out int insertedID)) {
Console.WriteLine($"newly inserted ID: {insertedID}"); }
else { Console.WriteLine("Failed to retrieve the inserted ID!"); }

connection.Close();
} catch (Exception ex) { Console.WriteLine(ex.Message); }

}


static void Main(string[] args)
{
stContact contact = new stContact {
FirstName="Laila",
LastName="Maher",
Email="m@example.com",
Phone="1234567890",
Address="123 Main Street",
CountryID=1,
};

AddNewContactAndGetID(contact);

Console.ReadLine();
}
}
```

ال out اللي حطها في جمله ال if دي بدل ما يعمل متغير خاص بيها انما انت ممكن بعدين تعمله static في مكان تاني او حسب ماانت محتاج

Update Data

زي ماقولنا قبل كده انه عمليات الاضافه والتعديل والحذف بنستخدم فيهم ال execute non query

كل اللي هتعمله انك هتغير في ال query وهتزود parameter خاص بالشرط اللي هتعدل عليه

وتقدر تعمل update لاكتر من record في نفس الوقت

```
using System;

using System.Data.SqlClient;
using System.Runtime.CompilerServices;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";



public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}
static void UpdateContact(int ContactID,stContact newContact) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = @"Update Contacts set
FirstName=@FirstName,
LastName=@LastName,
Email=@Email,
Phone=@Phone,
Address=@Address,
CountryID=@CountryID
Where ContactID=@ContactID";

SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@FirstName",newContact.FirstName);
command.Parameters.AddWithValue("@LastName", newContact.LastName);
command.Parameters.AddWithValue("@Email", newContact.Email);
command.Parameters.AddWithValue("@Phone", newContact.Phone);
command.Parameters.AddWithValue("@Address", newContact.Address);
command.Parameters.AddWithValue("@CountryID", newContact.CountryID);
command.Parameters.AddWithValue("@ContactID",ContactID);

try {

connection.Open();
int result=command.ExecuteNonQuery();

if (result >0) {
Console.WriteLine($"Updated Successfully"); }
else { Console.WriteLine("Update Failed!"); }

connection.Close();
} catch (Exception ex) { Console.WriteLine(ex.Message); }

}


static void Main(string[] args)
{
stContact contact = new stContact {
FirstName="Mohammed",
LastName="Abu-Hadhoud",
Email="m@example.com",
Phone="1234567890",
Address="123 Main Street",
CountryID=1,
};

UpdateContact(1,contact);

Console.ReadLine();
}
}
```

Delete Data

هنحذف داتا وخلي بالك عشان لو ماحطيتش أي شرط هتلاقي نفسك بتسرح بعربية فول وتجيلك الازالة تعملك delete لحياتك

نفس الحوار هتعدل عال query وتستخدم non query

```
using System;

using System.Data.SqlClient;
using System.Runtime.CompilerServices;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";



public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}


static void DeleteContact(int ContactID) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = @"Delete from Contacts Where ContactID=@ContactID";
SqlCommand command = new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@ContactID",ContactID);

try {

connection.Open();
int result=command.ExecuteNonQuery();

if (result >0) {
Console.WriteLine($"Deleted Successfully"); }
else { Console.WriteLine("Delete Failed!"); }

connection.Close();
} catch (Exception ex) { Console.WriteLine(ex.Message); }

}


static void Main(string[] args)
{
DeleteContact(10);

Console.ReadLine();
}
}
```

Handle In Statement

كنا في الsql لما كنا بنضيف شرط معين ونستخدم ال or كان فيه حاجه اسهل منها و هيا انك تكتب in وتفتح قوسين تحط فيهم كل الخيارات دي اللي بيتكلم عنها

هنا هيديك مثال عال delete انه هيحذف اكتر من record واللي هتعمله هنا هتعمله في أي حاجه تانيه

الفكره هنا انك هتخلي ال function بدل كانت بتاخد int لا هتخليها تاخد string واليوزر هيحط القيم ك string بين كل قيمه و التانيه فاصله

طب وال query ؟

قالك ماينفعش تستخدم فيها ال parameterized query بتستخدم ال concatenation انك ترزع المتغير في نص ال query

```csharp
using System;

using System.Data.SqlClient;
using System.Runtime.CompilerServices;

internal class Program
{
static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456;";



public struct stContact {
public int ID { get; set; }
public string FirstName { get; set; }
public string LastName { get; set; }
public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public int CountryID { get; set; }
}


static void DeleteContacts(string ContactsID) {

SqlConnection connection=new SqlConnection(ConnectionString);

string Query = @"Delete from Contacts Where ContactID in("+ContactsID+")";
SqlCommand command = new SqlCommand(Query, connection);


try {

connection.Open();
int result=command.ExecuteNonQuery();

if (result >0) {
Console.WriteLine($"Deleted Successfully"); }
else { Console.WriteLine("Delete Failed!"); }

connection.Close();
} catch (Exception ex) { Console.WriteLine(ex.Message); }

}


static void Main(string[] args)
{
DeleteContacts("11,8,9");

Console.ReadLine();
}
}
```

<u>What are CRUD Operations?</u>

ال CRUD Operations وفيها ال CRUD اختصار ل Create - Read – Update – Delete

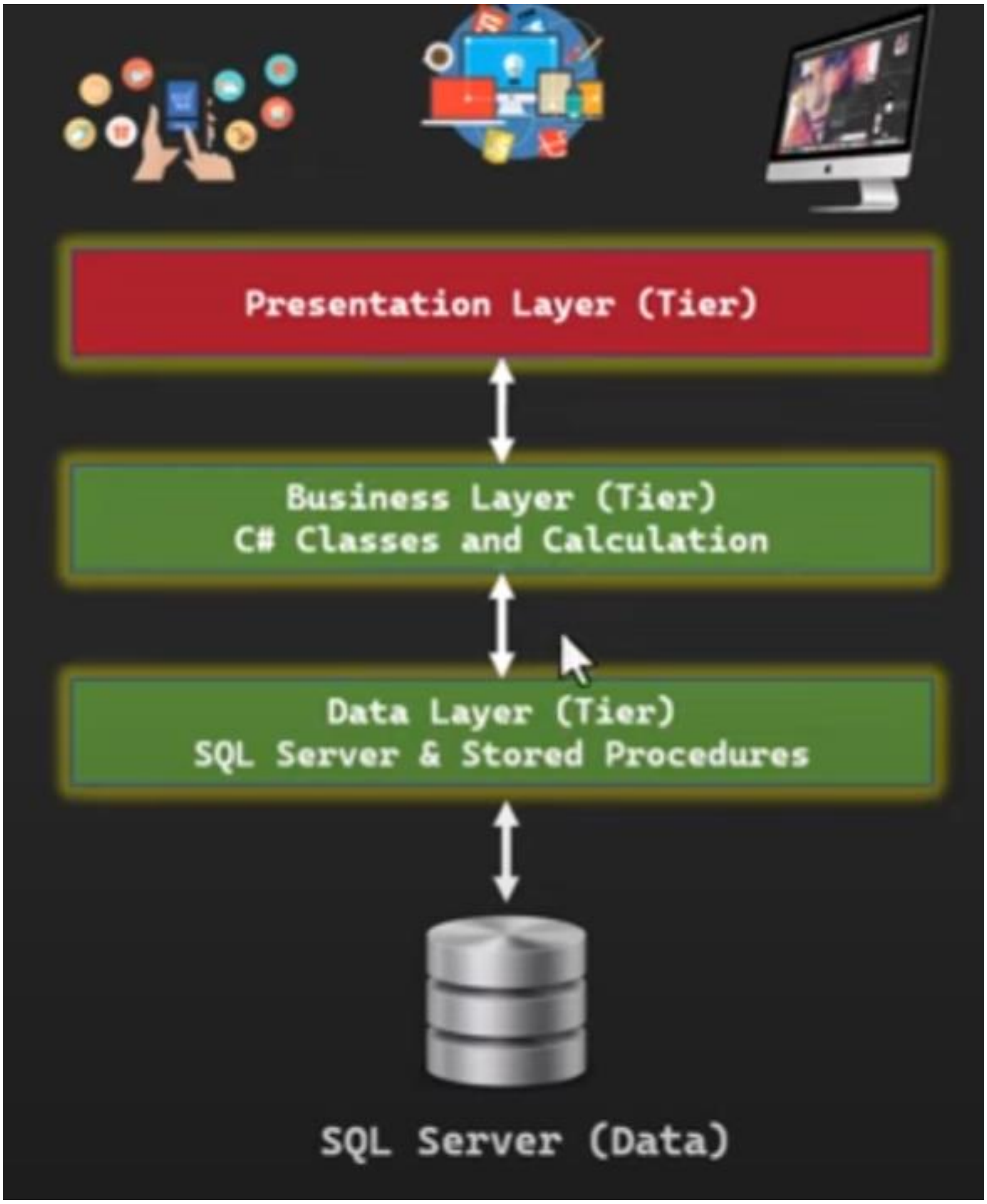
و المصطلح ده بيعبر عن كل العمليات اللي بتحتاجها لما بتتعامل مع الداتابيز

What is 3-Tier Architecture? and Why?

ال 3tier architecture وفكرته انك بتكتب الكود مره واحده وبتستخدمه اكتر من مره وعشان تعمل ده لازم تكتب clean code او سياسة فرق تسد

فال 3tier architecture هوا من اسمه مكون من 3 أجزاء :-

1- Presentation layer :- وهوا الجزء الخاص بالتصميم والواجهات وده بتكتب فيه الاكواد الخاصه بالواجهات فقط وده بو هتعمل 3 تطبيقات مثلا موبايل وويب وواحد للكمبيوتر
2- ال business layer :- وده الكود اللي فيه ال logic اللي هتستخدمه في تشغيل البرنامج بغض النظر عن اللغه اللي هتكتب بيها وفيه بتعدل او تضيف functions مره واحده بس وبتقدر تستخدم الاكواد اللي فيه في أي تطبيق من التطبيقات المرتبطه بيه
3- Database layer وده الخاص بالتواصل مع الداتابيز


Presentation Layer (Tier)
Business Layer (Tier)
C# Classes and Calculation
Data Layer (Tier)
SQL Server & Stored Procedures
SQL Server (Data)

Contacts Solution (Console app) - Find Contact

هنعمل مشروع ال contacts باستخدام ال tier3 فال presentation layer مره هيكون console ومرة هيكون windows forms

وال 2 layers التانيين هيمثلوا مكتبات لينا بمعني اننا هنستخدمهم في المشروعين دول واي مشروع تاني

اول حاجه هنعمل restore للداتابيز

```
use ContactsDB
EXEC sp_changedbowner 'sa';
```

طيب هنفكر في المشروع ازاي؟

اول حاجه احنا عاوزين نعمل 3tier architecture فاانت انسي كلمة 3 tier دي واستدبلها ب 3 classes يعني هنعمل كلاس لكل tier

تاني حاجه هنفكر في المشروع علي انه مكون من مجموعه من ال properties او الخصائص فهنمسك كل خاصيه نشتغل عليها لوحدها في كل كلاس

ثالث حاجه وهيا انه ال presentation layer ده هيكون فيه كل الحاجات الخاصه باستعراض الداتا و function واحده بس او سطر كود واحد هيجيبلك الداتا والداتا اللي هتجيلك لازم تكون جاهزه علي العرض (هنا بتكلم عموما مش المشروع ده بس)

ال business layer ده هيكون الوسيط بين ال presentation layer وال data access layer وده هتعالج فيه الداتا اللي جايالك قبل ما تطلعها لل presentation layer

ال data access layer وده امين المخزن يقولك فين العربيه اللي هتحمل عليها البضاعه ويقوم داخل المخزن جايب البضاعه محملهالك

طيب تعالي نجهز ال layers بتاعت المشروع

اول حاجه هنعمل مشروع جديد من نوع console app

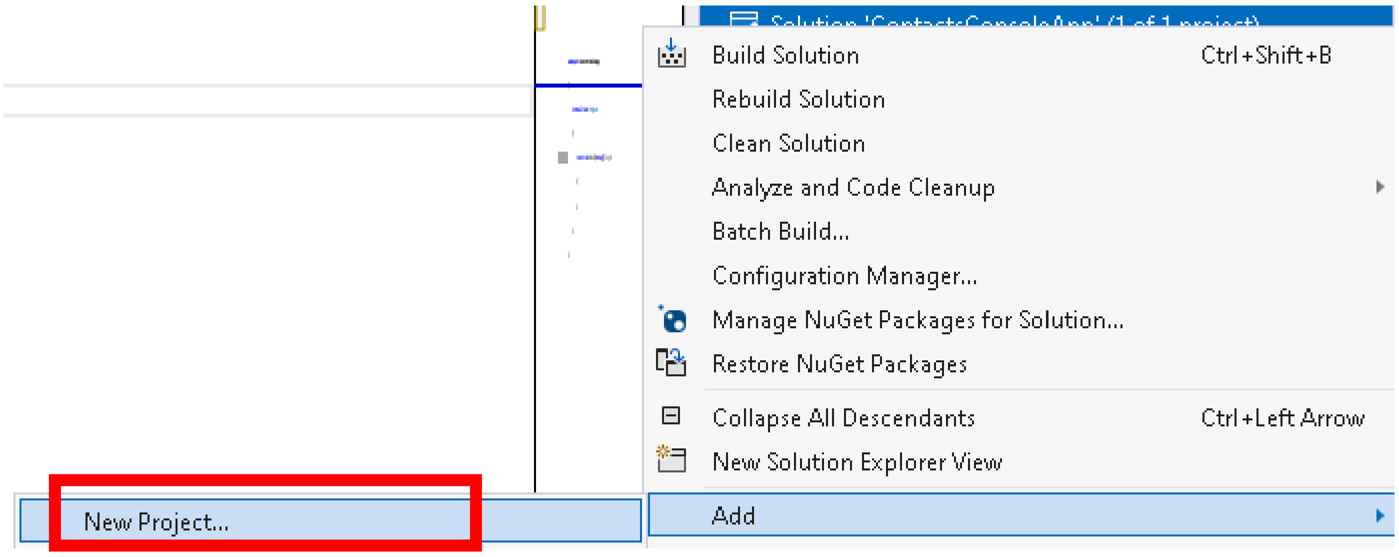
Project name

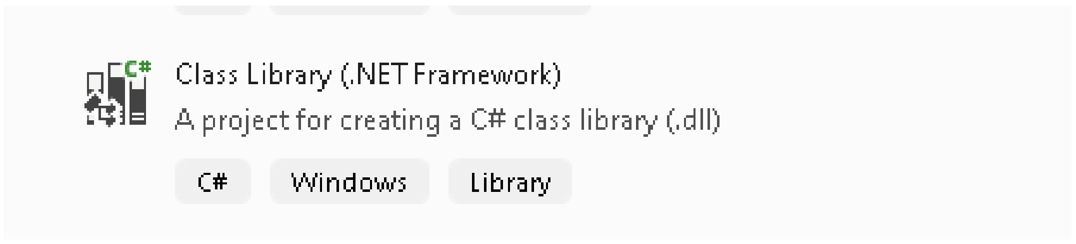
ContactsConsoleApp

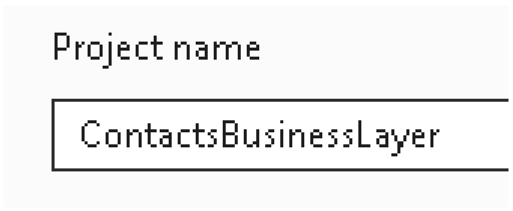
```
using System;

namespace ContactsConsoleApp
{
    internal class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

وبكده يبقي جهزنا ال presentation layer

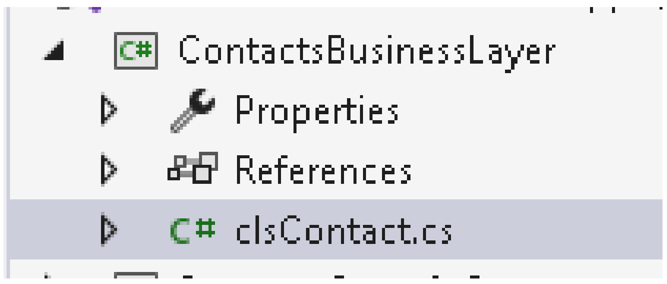
هنجهز ال business layer كليك يمين علي ال solution وبعدين add وبعدين new project


Solution 'ContactsConsoleApp' (1 of 1 project)
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Analyze and Code Cleanup
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
Restore NuGet Packages
Collapse All Descendants Ctrl+Left Arrow
New Solution Explorer View
Add
New Project...


Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C#
Windows
Library

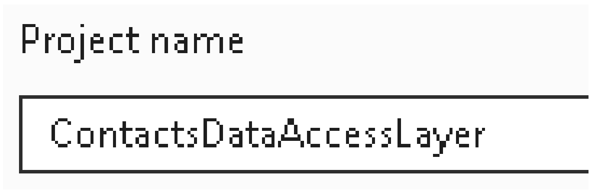


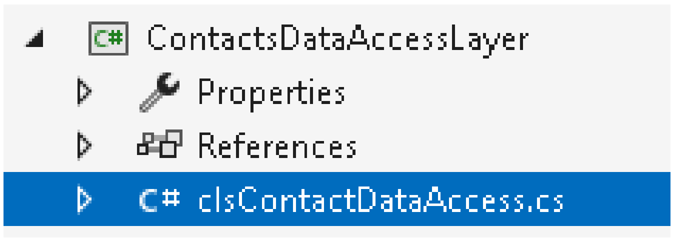
وبعدين اعمل rename للكلاس اللي جواه



```
using System;

namespace ContactsBusinessLayer
{
public class clsContact
{
}
}
```

هنعمل ال data access layer بنفس الطريقه

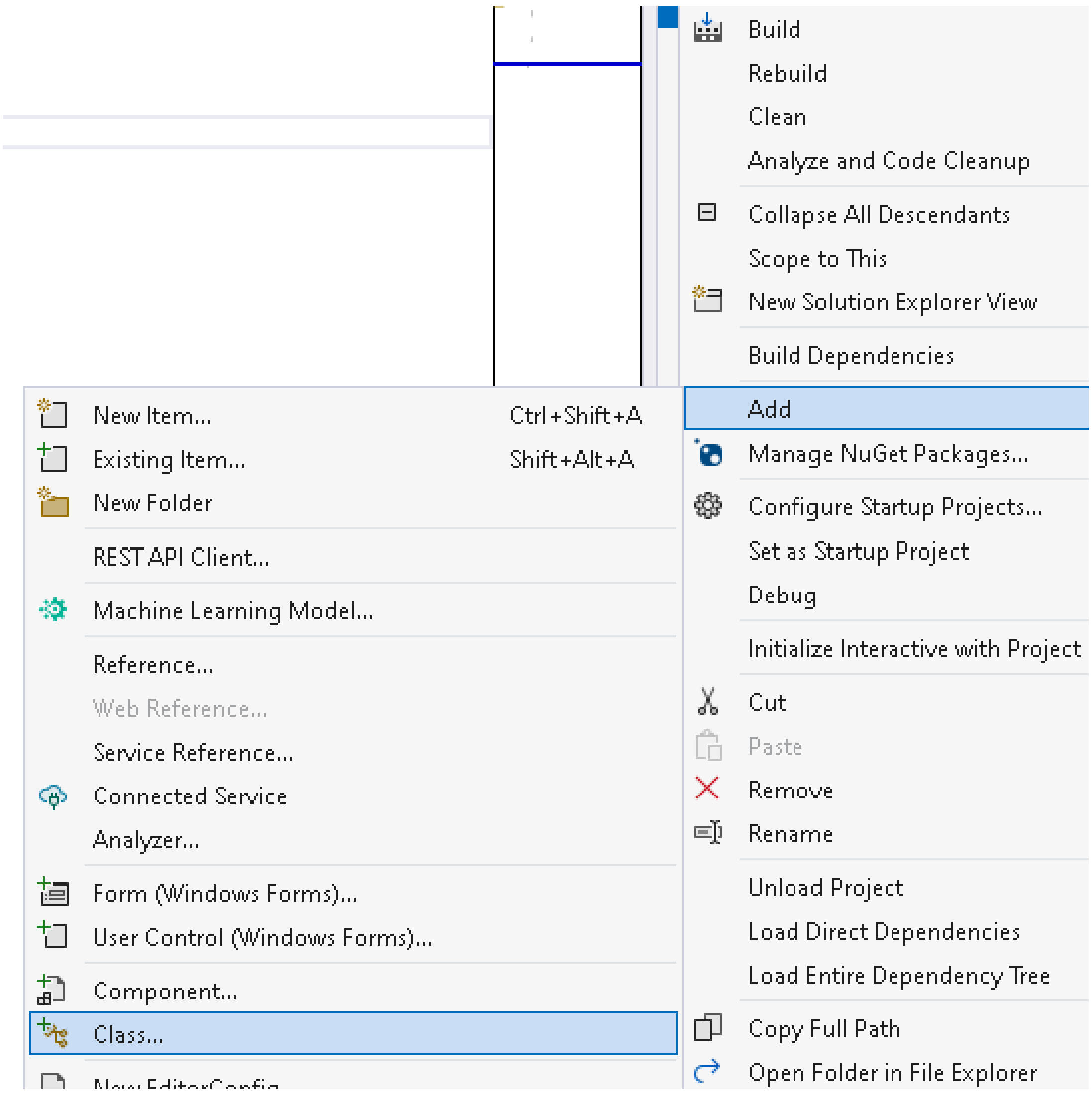


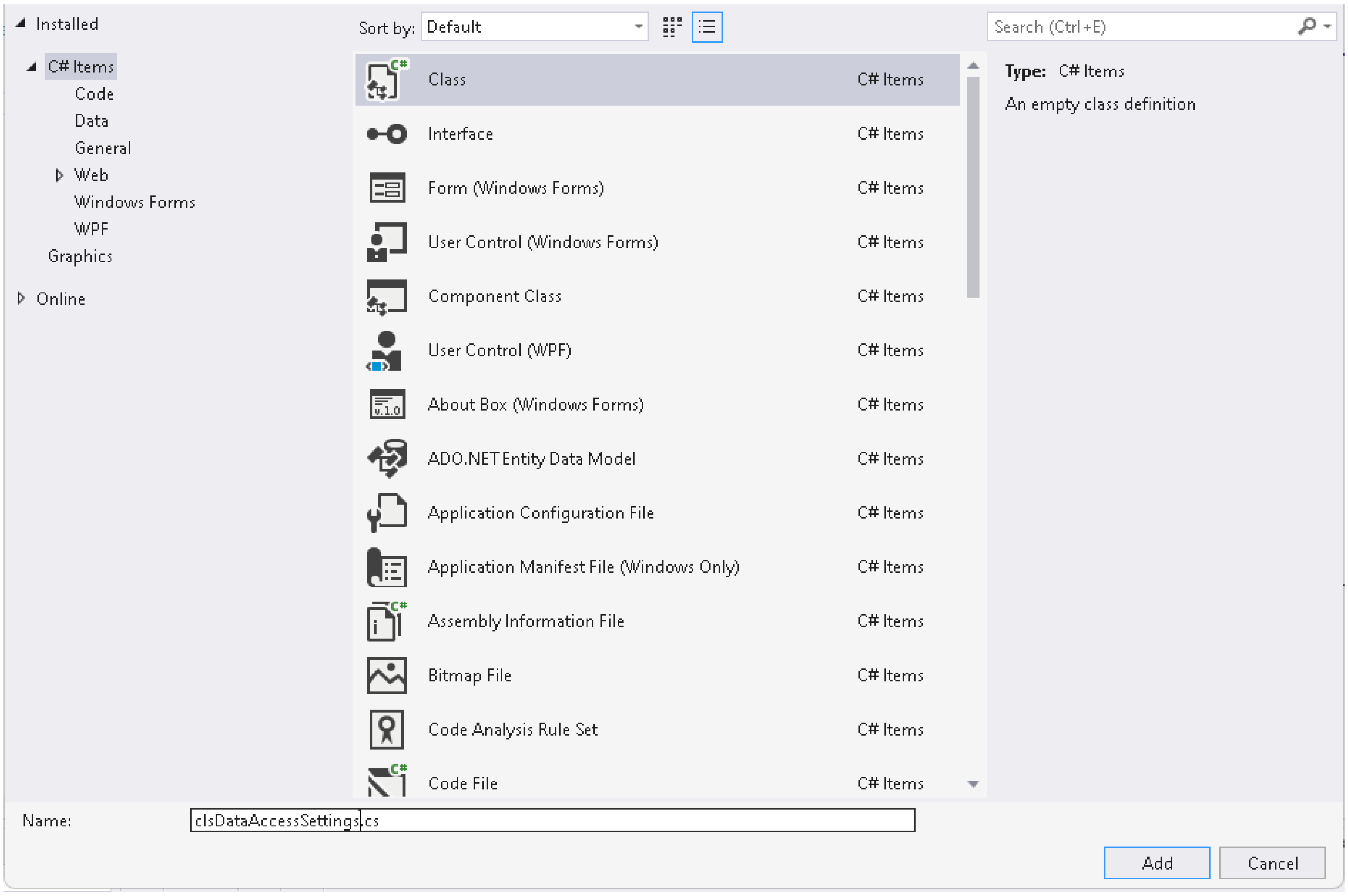


```
using System;

namespace ContactsDataAccessLayer
{
public class clsContactDataAccess
{
}
}
```

جوه ال layer ده هنضيف كلاس تاني هنحط فيه ال connection string

كليك يمين عال ContactsDataAccessLayer وبعدين add وبعدين new class



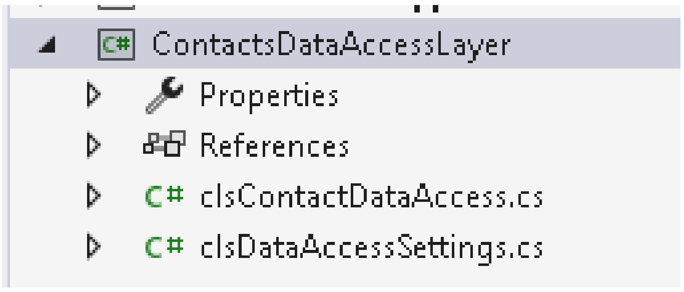


Installed
C# Items
Code
Data
General
Web
Windows Forms
WPF
Graphics
Online
Sort by: Default
Search (Ctrl+E)
Class C# Items
Interface C# Items
Form (Windows Forms) C# Items
User Control (Windows Forms) C# Items
Component Class C# Items
User Control (WPF) C# Items
About Box (Windows Forms) C# Items
ADO.NET Entity Data Model C# Items
Application Configuration File C# Items
Application Manifest File (Windows Only) C# Items
Assembly Information File C# Items
Bitmap File C# Items
Code Analysis Rule Set C# Items
Code File C# Items
Type: C# Items
An empty class definition
Add
Cancel

Name: clsDataAccessSettings.cs

```
using System;

namespace ContactsDataAccessLayer
{
static class clsDataAccessSettings
{
}
}
```

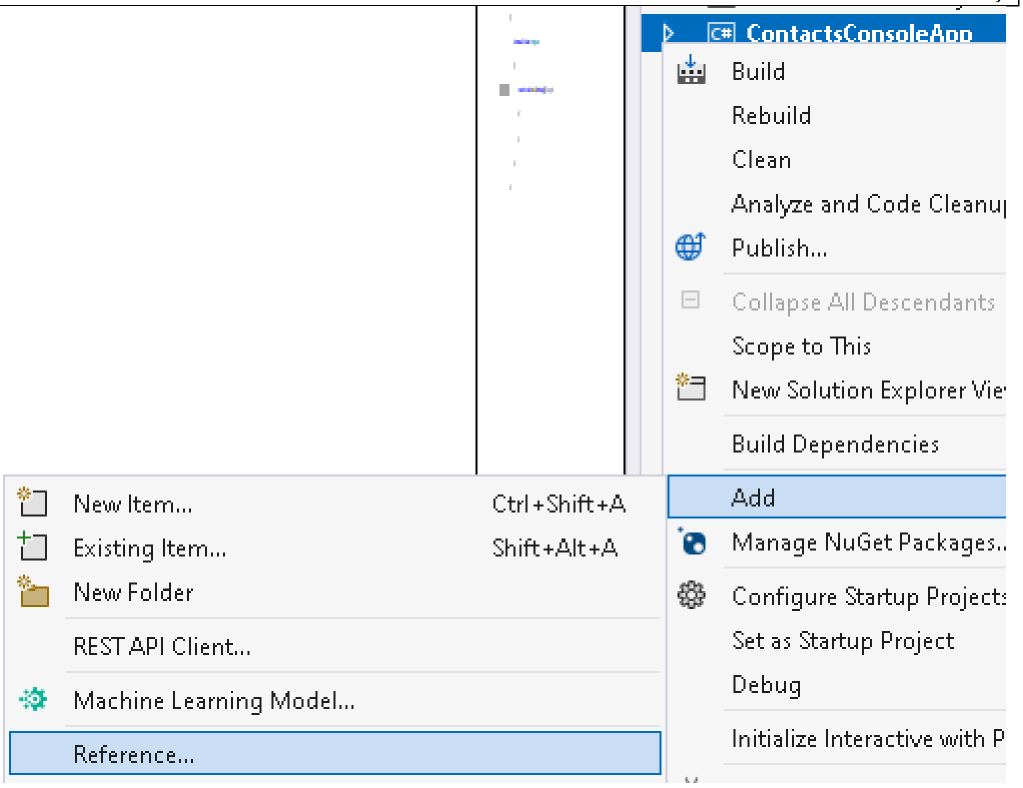
خليناه static

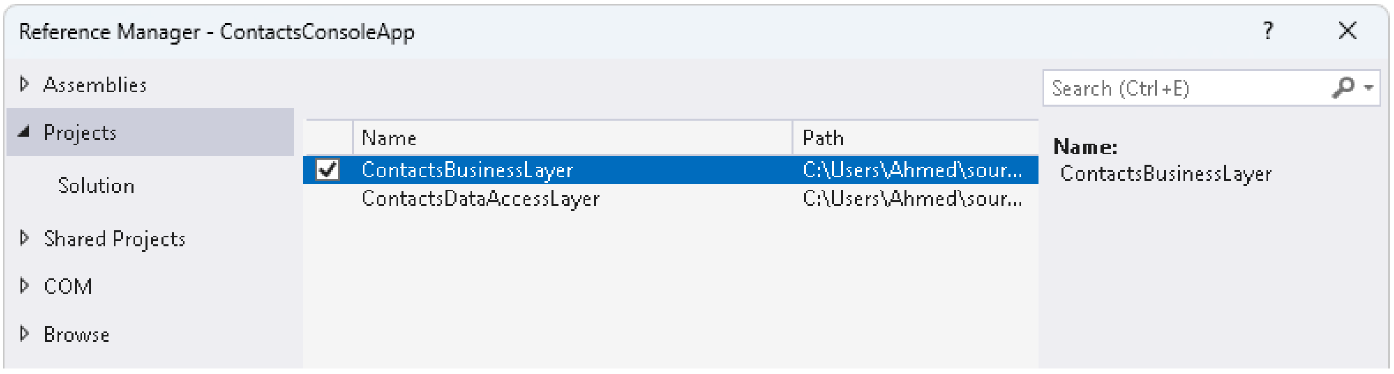


بعد كد هنيجي عند ال console app ونضيف فيه ال business layer ونعمله reference

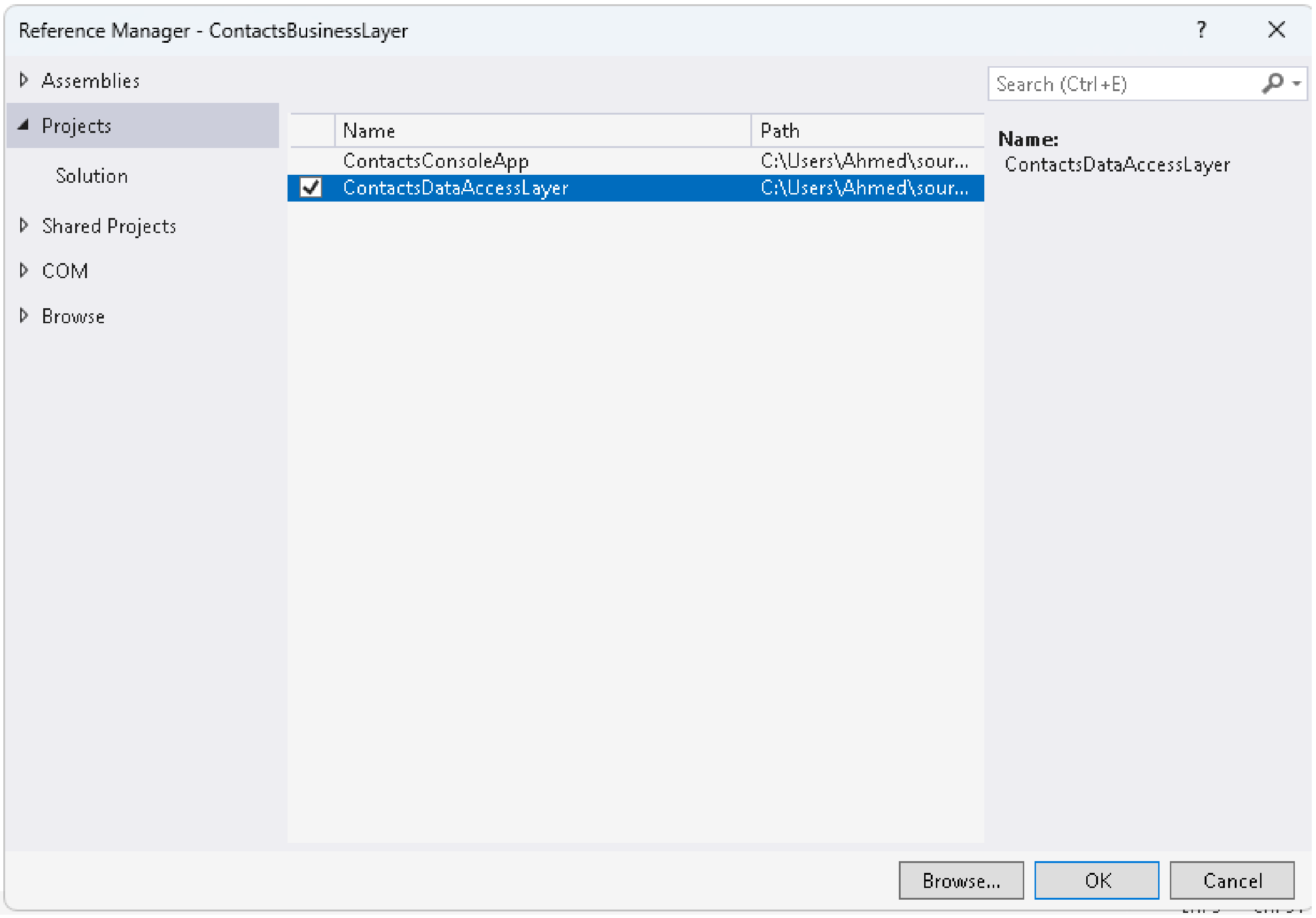
```
using System;
using ContactsBusinessLayer;

namespace ContactsConsoleApp
{
internal class Program
{
static void Main(string[] args)
{
}
```

```
                                                                                                    }
                                                                                                    }
```




Reference Manager - ContactsConsoleApp
Assemblies
Projects
Solution
Shared Projects
COM
Browse
Search (Ctrl+E)
Name
Path
ContactsBusinessLayer C:\Users\Ahmed\sour...
ContactsDataAccessLayer C:\Users\Ahmed\sour...
Name:
ContactsBusinessLayer

و هنيجي لل business layer ونعمل reference لل data access بنفس الطريقه



	Name	Path
	ContactsConsoleApp	C:\Users\Ahmed\sour...
✓	ContactsDataAccessLayer	C:\Users\Ahmed\sour...

Name:
ContactsDataAccessLayer

```
using System;

using ContactsDataAccessLayer;
namespace ContactsBusinessLayer
{
public class clsContact
{
}
}
```

كده جهزنا الملفات اللي هنكتب فيها

تعالي ندخل عالمشروع واول خاصيه عاوزين نضيفها فيه اننا نبحث عن contact عن طريق ال id

اول وابسط خطوه تعالي نروح لكلاس ال clsDataAccessSettings ونحط فيه ال connection string

```
using System;


namespace ContactsDataAccessLayer
{
static class clsDataAccessSettings
{
public static string ConnectionString = "Server=.;Database=ContactsDB;User Id=sa;Password=sa123456";
}
}
```

تعالي بقي للشغل

هنبدأ نبني البرنامج من تحت لفوق يعني هنبدأ من عند data access layer

اللي هيحصل في ال layer ده اننا هنعمل methods بتاخذ parameters ونعبي فيها الداتا

ملحوظه بالنسب هلل image path عملنا عليها ifstatement لانها في الداتابيز بتقبل null فلازم تهندلها

احنا هنا جوه ال data access layer

ده الكلاس

عاملين ال method من النوع static عشان ماضطرش اخد object من الكلاس وباخد parameters عشان اعبيها by ref

ده متغير عشان لو حصل خطأ اعرف منه

كل ده عادي بقي بعمل connection و query و reader وبعبي الداتا

هنا بيقولك لو كنت قفلت ال connection بعد ماقفلت ال reader كان ممكن يحصل exception عند سطر معين بعد مايفتح ال connection وساعتها هيفضل ال connection مفتوح لانه مش يكمل و هينط علي طول عال exception فعشان كده حط السطر بتاع قفل ال connection في حاجه اسمها finally وده الكود اللي هيتنفذ سواء حصل exception او ماحصلش
وده الأفضل طبعا

```
using System;
using System.Data.SqlClient;


namespace ContactsDataAccessLayer
{
public class clsContactDataAccess
{
public static bool GetContactInfoByID(int ID,ref string FirstName,ref
string LastName,ref string Email,
ref string Phone ,ref string Address,ref DateTime DateOfBirth,ref int
CountryID,ref string ImagePath) {

bool IsFound = false;

SqlConnection connection=new
SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "select * from contacts where
ContactID=@ContactID";

SqlCommand command=new SqlCommand(Query, connection);
command.Parameters.AddWithValue("@ContactID", ID);

try {

connection.Open();
SqlDataReader Reader=command.ExecuteReader();

if (Reader.Read())
{
IsFound = true;

FirstName = (string)Reader["FirstName"];
LastName = (string)Reader["LastName"];
Email = (string)Reader["Email"];
Phone = (string)Reader["Phone"];
Address = (string)Reader["Address"];
DateOfBirth = (DateTime)Reader["DateOfBirth"];
CountryID = (int)Reader["CountryID"];

if (Reader["ImagePath"] !=DBNull.Value) {
ImagePath = (string)Reader["ImagePath"];
} else { ImagePath = ""; }
}
else { IsFound = false; }
Reader.Close();

} catch (Exception ex) {
IsFound = false;
} finally {
connection.Close();
}

return IsFound;

}
}
}
```

تمام نروح بقي لل business layer وده هيستقبل الداتا اللي جايه سواء من ال presentation او من ال data access ويبعتها للطرف التاني يعني شغال موصلاتي او عصفورة

اول حاجه هنعملها اننا هنعرف المتغيرات اللي هنستخدمها في تخزين الداتا عشان نعرف ننقلها

```
using System;

using ContactsDataAccessLayer;
namespace ContactsBusinessLayer
{
public class clsContact
{
public int ID { set; get; }
public string FirstName { set; get; }
public string LastName { set; get; }
public string Email { set; get; }
public string Phone { set; get; }
public string Address { set; get;}
public DateTime DateOfBirth { set; get; }
public string ImagePath {  set; get; }
public int CountryID {  set; get; }


}
}
```

بعد كده هنعمل اتنين constructor واحد هياخد parameters وده هيكون private وهيكون مخصص لنقل الداتا من ال data access layer لل presentation layer

والتاني هيكون public وهنعمل فيه initialization للمتغيرات بحيث اني لما اعمل object عن طريقه ال object ده هيكون فاضي وده هيكون خاص بنقل الداتا من ال presentation layer لل data access layer

```
public clsContact() {
this.ID = -1;
this.FirstName = "";
this.LastName = "";
this.Email = "";
this.Phone = "";
this.Address = "";
this.DateOfBirth=DateTime.Now;
this.CountryID = -1;
this.ImagePath = "";
}

private clsContact(int ID, string FirstName, string LastName, string Email,
string Phone, string Address, DateTime DateOfBirth, string ImagePath, int CountryID) {

this.ID = ID;
this.FirstName = FirstName;
this.LastName = LastName;
this.Email = Email;
this.Phone = Phone;
this.Address = Address;
this.DateOfBirth = DateOfBirth;
this.ImagePath = ImagePath;
this.CountryID = CountryID;
}
```

بعدين نعمل ال function بتاعت ال find وفيها هعرف متغيرات وامررهم لل function اللي عملناها في ال data access layer واخدهم ارجع بيهم object

```
public static clsContact Find(int ID) {

string FirstName = "", LastName = "", Email = "", Phone = "", Address = "", ImagePath = "";
DateTime DateOfBirth = DateTime.Now;
int CountryID = -1;

if (clsContactDataAccess.GetContactInfoByID(ID,ref FirstName,ref LastName,ref Email,
ref Phone,ref Address,ref DateOfBirth,ref CountryID,ref ImagePath)) {

return new clsContact(ID,FirstName,LastName,Email,Phone,Address,DateOfBirth,ImagePath,CountryID);

}else { return null; }

}
```

ندخل بقي عال presentation layer وفيه هنحط كل حاجه خاصه بالواجهه وفيه يدوب هاخد object من الكلاس clsContact عن طريق استدعاء ال function اللي اسمها find

```
using System;
using ContactsBusinessLayer;

namespace ContactsConsoleApp
{
internal class Program
{
static void testFindContact(int ID) {
clsContact Contact1 = clsContact.Find(ID);

if (Contact1 != null) {

Console.WriteLine(Contact1.FirstName+" "+Contact1.LastName);
Console.WriteLine(Contact1.Email);
Console.WriteLine(Contact1.Phone);
Console.WriteLine(Contact1.Address);
Console.WriteLine(Contact1.DateOfBirth);
Console.WriteLine(Contact1.CountryID);
Console.WriteLine(Contact1.ImagePath);
}
else {

Console.WriteLine("Contact [" + ID + "] Not Found!");
}
}


static void Main(string[] args)
{
testFindContact(1);
Console.ReadKey();
}
}
}
```

Add New Contact

هنزود خاصيه اننا نضيف contact في الداتا بيز

نبدأ بال data access layer

برضه هتاخد منك متغيرات بس المرادي فيها داتا وهتروح توديها للداتابيز وعاوزين واحنا بنضيف الداتا نرجع ال id اللي اتعمل

```
public static int AddNewContact(string FirstName, string LastName, string Email, string Phone,
string Address, DateTime DateOfBirth, int CountryID, string ImagePath) {

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);
string Query = @"insert into Contacts (FirstName,LastName,Email,Phone,Address,DateOfBirth,CountryID,ImagePath)
values (@FirstName,@LastName,@Email,@Phone,@Address,@DateOfBirth,@CountryID,@ImagePath);
select SCOPE_IDENTITY();";

SqlCommand Command = new SqlCommand(Query, connection);
Command.Parameters.AddWithValue("@FirstName", FirstName);
Command.Parameters.AddWithValue("@LastName", LastName);
Command.Parameters.AddWithValue("@Email", Email);
Command.Parameters.AddWithValue("@Phone", Phone);
Command.Parameters.AddWithValue("@Address", Address);
Command.Parameters.AddWithValue("@DateOfBirth", DateOfBirth);
Command.Parameters.AddWithValue("@CountryID", CountryID);

if (ImagePath !="") {
Command.Parameters.AddWithValue("@ImagePath", ImagePath);
} else {
Command.Parameters.AddWithValue("@ImagePath", System.DBNull.Value);
}

try {

connection.Open();

object result= Command.ExecuteScalar();

if (result !=null && int.TryParse(result.ToString(),out int InsertedID)) {

return InsertedID;
} else {
return -1;
}
} catch (Exception ex) { }finally { connection.Close(); }
return -1;
}
```

نيجي لل business layer

او حاجه هنضيف enum هيكون ال mode وده عشان يسهل علينا عمليات الاضافه والتعديلوفي بداية الكلاس هنعمله addnew وهنغير فيه من جوه ال methods بعدين

```
public class clsContact
{
public enum enMode {AddNew=0,Update=1}
public enMode Mode = enMode.AddNew;
```

طيب احنا دلوقتي عندنا اتنين constructor واحد بيجيب الداتا من presentation layer ونوعه public وطالما جايبلي داتا من طرف اليوزر يبقي اكيد عايز يعمل اضافه للداتابيز

يبقي هغير ال mode فيه اخليه addnew

وال constructor التاني بيجيب الداتا من طرف الداتابيز يبقي اكيد عايز اعدل عليها يبقي هغير ال mode في واخليه update

```
public clsContact() {
this.ID = -1;
this.FirstName = "";
this.LastName = "";
this.Email = "";
this.Phone = "";
this.Address = "";
this.DateOfBirth=DateTime.Now;
this.CountryID = -1;
this.ImagePath = "";
this.Mode = enMode.AddNew;
}

private clsContact(int ID, string FirstName, string LastName, string Email,
string Phone, string Address, DateTime DateOfBirth, string ImagePath, int CountryID) {

this.ID = ID;
this.FirstName = FirstName;
this.LastName = LastName;
this.Email = Email;
this.Phone = Phone;
this.Address = Address;
this.DateOfBirth = DateOfBirth;
this.ImagePath = ImagePath;
this.CountryID = CountryID;
this.Mode = enMode.Update;
}
```

بكده لما اجي استدعي ال METHOD اللي اسمها find هترجعلي object معمول عن طريق ال constructor اللي بيخلي ال mode يبقي update

طيب تعالي بقي نعمل ال function بتاعت ال addnew ودي كل اللي هتعمله انها هتستدعي ال method اللي عملناها في ال data access layer و هتقولي ان كانت النتيجه -1 ولا لا

بس عاوزين لما نروح لل presentation layer نستدعيها من method تانيه هنسميها save يبقي هنخليها private مش public

```
private bool _AddnewContact() {
this.ID = clsContactDataAccess.AddNewContact(this.FirstName,this.LastName,this.Email,this.Phone,
this.Address,this.DateOfBirth,this.CountryID,this.ImagePath);

return (this.ID !=-1);
}
```

وبعدين نعمل ال function اللي اسمها save وفيها هنقول لو كان ال mode بيساوي add new يبقي هستدعي ال method بتاعت ال add new

وبعد ماتتم الاضافه الكلاس هيكون فيه لسه معلومات ال contact زي ماهيا فلو عايز اعمل حفظ تاني هنا مش هضيف جديد هنا هعدل بس عشان كده هغير ال mode واخليه update

```
public bool Save() {
switch (Mode) {
case enMode.AddNew:
if (_AddnewContact()) {
Mode = enMode.Update;
return true;
} else { return false; }

case enMode.Update:
break;


}
return false;
}
```

اخر حاجه ال presentation layer

```
static void testAddNewContact() {
clsContact Contact1=new clsContact();
Contact1.FirstName = "Fadi";
Contact1.LastName = "Maher";
Contact1.Email = "A@a.com";
Contact1.Phone = "010010";
Contact1.Address = "address1";
Contact1.DateOfBirth =new DateTime(1977,11,6,10,30,0) ;
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save()) {
Console.WriteLine("Contact Added Successfully with Id=" + Contact1.ID);
} else {
Console.WriteLine("Error");
}

}
```

```
static void Main(string[] args)
{
// testFindContact(1);
testAddNewContact();
Console.ReadKey();
}
```

Update Contact

يلا نعمل كود ال update

ال data access layer

```
public static bool UpdateContact(int ID,string FirstName,string LastName,string Email,
string Phone,string Address,DateTime DateOfBirth,int CountryID,string ImagePath) {

int RowsAffected = 0;

SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = @"Update Contacts set FirstName=@FirstName,LastName=@LastName,
Email=@Email,Phone=@Phone,Address=@Address,DateOfBirth=@DateOfBirth,
CountryID=@CountryID,ImagePath=@ImagePath
where ContactID=@ContactID";

SqlCommand command=new SqlCommand(Query,Connection);

command.Parameters.AddWithValue("@ContactID", ID);
command.Parameters.AddWithValue ("@FirstName", FirstName);
command.Parameters.AddWithValue ("@LastName", LastName);
command.Parameters.AddWithValue ("@Email", Email);
command.Parameters.AddWithValue ("@Phone", Phone);
command.Parameters.AddWithValue ("@Address", Address);
command.Parameters.AddWithValue ("@DateOfBirth", DateOfBirth);
command.Parameters.AddWithValue ("@CountryID", CountryID);

if (ImagePath !="") {
command.Parameters.AddWithValue ("@ImagePath", ImagePath);
} else {
command.Parameters.AddWithValue("@ImagePath", System.DBNull.Value);
}

try {

Connection.Open();
RowsAffected = command.ExecuteNonQuery();

}catch (Exception ex) { return false; }finally {  Connection.Close(); }


return (RowsAffected>0);
}
```

وده اللي هنزوده في ال business layer

```
private bool _UpdateContact() {

return clsContactDataAccess.UpdateContact(this.ID,this.FirstName,this.LastName,this.Email,this.Phone,
this.Address,this.DateOfBirth,this.CountryID,this.ImagePath);

}

public bool Save() {
switch (Mode) {
case enMode.AddNew:
if (_AddnewContact()) {
Mode = enMode.Update;
return true;
} else { return false; }

case enMode.Update:
return (_UpdateContact());
}
return false;
}
```

وده اللي هنزوده في ال presentation layer

```
static void testUpdateContact(int ID) {

clsContact Contact1 = clsContact.Find(ID);
Contact1.FirstName = "Lina";
Contact1.LastName = "Maher";
Contact1.Email = "A2@a.com";
Contact1.Phone = "2222";
Contact1.Address = "222";
Contact1.DateOfBirth = new DateTime(1977, 11, 6, 10, 30, 0);
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save()) {
Console.WriteLine("Contact Updated Successfuly");
} else { Console.WriteLine("Contact with ID="+ID+" not found!"); }

}
```

Delete Contact

ال delete ودي اسهل حاجه وبيقولك مش لازم تعمل find وتعبي object ادخ عال query علي طول واحذف

ده اللي زودناه في ال data access

```
public static bool DeleteContact(int ContactID) {

int RowsAffected = 0;

SqlConnection Connection= new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = @"delete Contacts where ContactID=@ContactID";

SqlCommand Command=new SqlCommand(Query,Connection);

Command.Parameters.AddWithValue("@ContactID", ContactID);

try {
Connection.Open ();
RowsAffected=Command.ExecuteNonQuery();

} catch (Exception ex) { }finally { Connection.Close(); }
return (RowsAffected>0);
}
```

وده اللي زودناه في ال business

```
public static bool DeleteContact(int ID) {
return clsContactDataAccess.DeleteContact(ID);
}
```

وده ال presentation

```
static void testDeleteContact(int ID)
{
if (clsContact.DeleteContact(ID))
{

Console.WriteLine("DELETED SUCCESSFULLY");
}
else { Console.WriteLine("DELETE FAILED"); }

}
```

List Contacts

عاوزين دلوقتي نجيب الداتا اللي في الجدول كله اول حاجه هتفكر فيها هيا اني هستخدم ال data reader عشان اجيب الداتا من الجدول وهمشي عليهم ب while loop عشان اعبيهم

طيب تاني حاجه وهيا اني محتاج dynamic array ويكون two dimentional كمان عشان اقدر اعبي فيه الداتا واتحكم فيها بحريه

هنا قالك فيه حاجه اسمها data table دي هنيجي لتفاصيلها بعدين لان موضوعها كبير فااحنا هنستخدم اللي احنا عاوزينه منها دلوقتي

طيب ال data reader اللي هيجيبلي الداتا من الجدول فيه حاجه بترجع true و false واسمها has rows ودي بتقولك الجدول اللي راجع ده فيه صفوف ولا لا يعني فيه داتا ولا لا

وعندك ال data table اللي هنعبي فيه الداتا فيه function اسمها load ودي لو فيه جدول عندك بتاخده وتعبيه زي ماهوا في ال object اللي اخدته وبدون أي مجهود (الله يرحم ال c++)

فااحنا اللي هنعمله هناخد ال reader ونحمله عال data table وبارك الله فيما رزق وكان الله بالسر عليم

طيب لما نيجي نعرض الداتا في شاشة ال console هنعمل ايه؟

فيه في ال data table حاجه اسمها rows ودي عباره عن الصفوف بتاعت الجدول

وفيه كلاس اسمه data row وده عباره عن صف

بص اعتبر ال data table ده عباره عن two dimentional array والصفوف اللي فيه اسمها rows
وال data row اعتبره one dimentional array

فاالحنا هنلف علي صف صف ب for loop وناخد منه الداتا اللي عاوزينها ونطبعها

يلا بينا نشوف ازاي؟

اول حاجه ال data access layer

```
public static DataTable GetAllContacts() {

DataTable dataTable=new DataTable();

SqlConnection Connection=new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "select * from Contacts";

SqlCommand Command= new SqlCommand(Query,Connection);

try {

Connection.Open();
SqlDataReader reader=Command.ExecuteReader();

if (reader.HasRows) {
dataTable.Load(reader);
}
reader.Close();
Connection.Close();
}catch (Exception ex) { }finally { Connection.Close(); }
return dataTable;
}
```

بعدين ال presentation layer

```
public static DataTable GetAllContacts() {
return clsContactDataAccess.GetAllContacts();

}
```

بعدين ال presentation

```
static void ListContacts() {

DataTable dataTable = clsContact.GetAllContacts();

foreach (DataRow row in dataTable.Rows)
{
Console.WriteLine($"{row["ContactID"]},  {row["FirstName"]} {row["LastName"]}");

}
}
```

Is Contact Exist

لو عايز تعرف ال contact الفلاني موجود ولا لا ممكن تعمل كده من غير ماتستعمل ال find وتعبي ال object بداتا عن طريق ال query دي

```
SELECT Found=1 FROM Contacts
where ContactID=1
```

يلا بينا نعملها في البرنامج بتاعنا وبعدين نعدل علي ال function بتاعت ال delete بحيث اننا نشوف ال contact موجوده ولا لا الأول قبل مانحذف

ده ال data acces layer

```
public static bool IsContactExist(int ID)
{
bool isFound = false;

SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);
string Query = "SELECT Found=1 FROM Contacts where ContactID=@ContactID";

SqlCommand command = new SqlCommand(Query, Connection);
command.Parameters.AddWithValue("@ContactID", ID);
try
{

Connection.Open();
SqlDataReader reader = command.ExecuteReader();

isFound = reader.HasRows;
reader.Close();
}
catch (Exception ex) { isFound = false; }
finally { Connection.Close(); }
return isFound;
}
```

ده ال business

```
public static bool isContactExist(int ID) { return clsContactDataAccess.IsContactExist(ID); }
```

ده ال presentation

```
static void isContactExist(int ID) {
if (clsContact.isContactExist(ID))
{
Console.WriteLine("YES");
}
else { Console.WriteLine("NO"); }

}
```

وده التعديل عال delete في ال presentation

```
static void testDeleteContact(int ID)
{
if (clsContact.isContactExist(ID) ){

if (clsContact.DeleteContact(ID))
{

Console.WriteLine("DELETED SUCCESSFULLY");
}
else { Console.WriteLine("DELETE FAILED"); }

}
else { Console.WriteLine("not exist"); }
}
```

Homework 1: Prepare Business/DataAccess for Countries.

Homework:

Extend the project and you need to prepare **Business** and **DataAccess** Layers for Countries, do all methods done for contacts, in addition you should be able to :

- Find Country By Name.
- Check Country Existance by name as well.

in the presentation layer prepare test functions for all.

شايف كل اللي عملناه في ال contacts اعمل زيه في ال countries بس في نفس المشروع يعني زود data و business layer بس عال countries وتزود خاصيتين انك تعمل find باسم الدوله وتشوف is exist بالاسم

Homework 1: Solution

ده ال dataAccess

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Runtime.Remoting.Messaging;

namespace CountriesDataAccessLayer
{
public class CountriesDataAccess
{
public static bool GetCountryInfoByID(int ID,ref string CountryName) {

bool isFound = false;


SqlConnection Connection=new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "Select * from Countries where CountryID=@CountryID";

SqlCommand Command = new SqlCommand(Query, Connection);

Command.Parameters.AddWithValue("@CountryID",ID);

try {

Connection.Open();
SqlDataReader Reader=Command.ExecuteReader();
```

```
if (Reader.Read())
{
isFound = true;
CountryName = (string) Reader["CountryName"];
} else {
isFound = false;
}
Reader.Close();

} catch (Exception ex) { isFound = false; }finally { Connection.Close(); }

return isFound;
}

public static bool GetCountryInfoByName(ref int ID,  string CountryName)
{

bool isFound = false;


SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "Select * from Countries where CountryName = @CountryName ";

SqlCommand Command = new SqlCommand(Query, Connection);

Command.Parameters.AddWithValue("@CountryName", CountryName);

try
{

Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

if (Reader.Read())
{
isFound = true;
CountryName = (string)Reader["CountryName"];
ID= (int)Reader["CountryID"];
}
else
{
isFound = false;
}
Reader.Close();

}
catch (Exception ex) { isFound = false; }
finally { Connection.Close(); }

return isFound;
}

public static int AddNewCountry(string CountryName) {


SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "insert into Countries values(@CountryName); select SCOPE_IDENTITY(); ";

SqlCommand Command = new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue("@CountryName",CountryName);

try {

Connection.Open();
object result=Command.ExecuteScalar();

if (result !=null && int.TryParse(result.ToString(),out int InsertedID )) {
```

```
return InsertedID;
} else {
return -1;
}
} catch (Exception ex) { return -1; } finally { Connection.Close(); }
return -1;
}

public static bool UpdateCountry(int ID,string CountryName) {

int RowsAffected = 0;


SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);
string Query = "Update Countries set CountryName=@CountryName where CountryID=@CountryID";

SqlCommand Command=new SqlCommand(Query, Connection);

Command.Parameters.AddWithValue("@CountryName", CountryName);
Command.Parameters.AddWithValue("@CountryID", ID);
try {

Connection.Open();
RowsAffected=Command.ExecuteNonQuery();

}catch(Exception ex) { }
finally { Connection.Close(); }
return (RowsAffected > 0);

}

public static bool DeleteCountry(int ID)
{
int RowsAffected = 0;
SqlConnection Connection=new SqlConnection(clsDataAccessSettings.ConnectionString);
string Query = "delete  Countries where CountryID=@CountryID";

SqlCommand Command=new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue("@CountryID",ID);

try {

Connection.Open() ;
RowsAffected=Command.ExecuteNonQuery();
}catch(Exception ex) { }finally { Connection.Close(); }
return (RowsAffected > 0);
}

public static DataTable GetAllCountries() {

DataTable dt = new DataTable();
SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string Query = "select * from Countries";

SqlCommand Command=new SqlCommand(Query, Connection);

try {
Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();

if (Reader.HasRows) {
dt.Load(Reader);
Reader.Close();
} else { return null; }
}catch(Exception ex) { return null; }finally { Connection.Close(); }
return dt;
}

public static bool isCountryExistByID(int ID) {
```

```
bool isFound = false;
SqlConnection Connection=new SqlConnection( clsDataAccessSettings.ConnectionString);
string Query = "select isFound=1 from Countries where CountryID=@CountryID";

SqlCommand Command= new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue ("@CountryID", ID);

try {

Connection.Open();
SqlDataReader Reader=Command.ExecuteReader();
isFound = Reader.HasRows;
Reader.Close();

} catch (Exception ex) { isFound= false; }
finally{ Connection.Close(); }
return isFound;
}

public static bool isCountryExistByName(string Name)
{

bool isFound = false;
SqlConnection Connection = new SqlConnection(clsDataAccessSettings.ConnectionString);
string Query = "select isFound=1 from Countries where CountryName =@CountryName";

SqlCommand Command = new SqlCommand(Query, Connection);
Command.Parameters.AddWithValue("@CountryName", Name);

try
{

Connection.Open();
SqlDataReader Reader = Command.ExecuteReader();
isFound = Reader.HasRows;
Reader.Close();

}
catch (Exception ex) { isFound = false; }
finally { Connection.Close(); }
return isFound;
}
}
}
```

وده ال business

```
using System;
using System.Data;
using CountriesDataAccessLayer;

namespace CountriesBusinessLayer
{
public class clsCountry
{
enum enMode {Add=0,Update=1 };

enMode Mode = enMode.Add;
public int CountryID { get; set; }
public string CountryName { get; set; }

public clsCountry()
{
this.CountryID = -1;
this.CountryName = "";
this.Mode= enMode.Add;
}

private clsCountry(int CountryID,string CountryName) {
```

```
this.CountryID = CountryID;
this.CountryName = CountryName;
this.Mode = enMode.Update;
}

public static clsCountry FindCountryByID(int CountryID) {

string CountryName = "";
if (CountriesDataAccess.GetCountryInfoByID(CountryID, ref CountryName))
{

return new clsCountry(CountryID, CountryName);
}
else { return null; }
}

public static clsCountry FindCountryByName(string CountryName)
{
int  CountryID = -1;
if (CountriesDataAccess.GetCountryInfoByName(ref CountryID, CountryName))
{
return new clsCountry(CountryID, CountryName);
}
else { return null; }
}

private bool AddNewCountry() {

this.CountryID = CountriesDataAccess.AddNewCountry(this.CountryName);

return (this.CountryID != -1);
}

private bool UpdateCountry() {
return CountriesDataAccess.UpdateCountry(this.CountryID, this.CountryName);
}

public bool Save() {
switch (this.Mode) {

case enMode.Add:
if (AddNewCountry()) {
this.Mode = enMode.Update;
return true;
} else { return false; }

case enMode.Update:
return UpdateCountry();
}
return false;
}

public static bool DeleteCountry(int ID) {
return CountriesDataAccessLayer.CountriesDataAccess.DeleteCountry(ID);
}

public static DataTable GetAllCountries() {

return CountriesDataAccess.GetAllCountries();
}

public static bool isCountryExistByName(string Name) {
return CountriesDataAccess.isCountryExistByName(Name);
}

public static bool isCountryExistByID(int ID)
{
return CountriesDataAccess.isCountryExistByID(ID);
}
}
```

```
}
```

وده ال presentation

```
using System;
using System.Data;
using System.Diagnostics.Eventing.Reader;
using ContactsBusinessLayer;
using CountriesBusinessLayer;

namespace ContactsConsoleApp
{
internal class Program
{
static void testFindContact(int ID) {
clsContact Contact1 = clsContact.Find(ID);

if (Contact1 != null) {

Console.WriteLine(Contact1.FirstName+" "+Contact1.LastName);
Console.WriteLine(Contact1.Email);
Console.WriteLine(Contact1.Phone);
Console.WriteLine(Contact1.Address);
Console.WriteLine(Contact1.DateOfBirth);
Console.WriteLine(Contact1.CountryID);
Console.WriteLine(Contact1.ImagePath);
}
else {

Console.WriteLine("Contact [" + ID + "] Not Found!");
}
}

static void testAddNewContact() {
clsContact Contact1=new clsContact();
Contact1.FirstName = "Fadi";
Contact1.LastName = "Maher";
Contact1.Email = "A@a.com";
Contact1.Phone = "010010";
Contact1.Address = "address1";
Contact1.DateOfBirth =new DateTime(1977,11,6,10,30,0) ;
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save()) {
Console.WriteLine("Contact Added Successfully with Id=" + Contact1.ID);
} else {
Console.WriteLine("Error");
}

}

static void testUpdateContact(int ID) {

clsContact Contact1 = clsContact.Find(ID);
Contact1.FirstName = "Lina";
Contact1.LastName = "Maher";
Contact1.Email = "A2@a.com";
Contact1.Phone = "2222";
Contact1.Address = "222";
Contact1.DateOfBirth = new DateTime(1977, 11, 6, 10, 30, 0);
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save()) {
Console.WriteLine("Contact Updated Successfuly");
} else { Console.WriteLine("Contact with ID="+ID+" not found!"); }
```

```
}

static void testDeleteContact(int ID)
{
if (clsContact.isContactExist(ID) ){

if (clsContact.DeleteContact(ID))
{

Console.WriteLine("DELETED SUCCESSFULLY");
}
else { Console.WriteLine("DELETE FAILED"); }

}
else { Console.WriteLine("not exist"); }
}


static void ListContacts() {

DataTable dataTable = clsContact.GetAllContacts();

foreach (DataRow row in dataTable.Rows)
{
Console.WriteLine($"{row["ContactID"]},  {row["FirstName"]} {row["LastName"]}");

}
}

static void isContactExist(int ID) {
if (clsContact.isContactExist(ID))
{
Console.WriteLine("YES");
}
else { Console.WriteLine("NO"); }

}


static void testFindCountryByID(int ID) {
clsCountry Country1=clsCountry.FindCountryByID(ID);

if (Country1 !=null) {
Console.WriteLine(Country1.CountryID);
Console.WriteLine(Country1.CountryName);
} else {
Console.WriteLine("Not Found!");
}
}

static void testFindCountryByName(string CountryName)
{
clsCountry Country1 = clsCountry.FindCountryByName(CountryName);

if (Country1 != null)
{
Console.WriteLine(Country1.CountryID);
Console.WriteLine(Country1.CountryName);
}
else
{
Console.WriteLine("Not Found!");
}
}

static void testAddNewCountry() {
clsCountry Country1=new clsCountry();
Country1.CountryName = "Jordan";
if (Country1.Save()) {
```

```
Console.WriteLine("done the ID is " + Country1.CountryID);
} else { Console.WriteLine("Error "); }
}

static void testUpdateCountry() {
clsCountry Country1 = clsCountry.FindCountryByName("eee");

Country1.CountryName = "Egypt";
if (Country1.Save()) {
Console.WriteLine("Done");
} else { Console.WriteLine("Error"); }
}

static void testDeleteCountry(int ID) {

if(clsCountry.isCountryExistByID(ID)){

if (clsCountry.DeleteCountry(ID))
{
Console.WriteLine("Done");
}
else { Console.WriteLine("Error"); }

}
else { Console.WriteLine("not found"); }

}

static void isCountryExistByID(int ID) {
if (clsCountry.isCountryExistByID(ID))
{

Console.WriteLine("yes");

}
else { Console.WriteLine("no"); }

}

static void isCountryExistByName(string Name)
{
if (clsCountry.isCountryExistByName(Name))
{

Console.WriteLine("yes");

}
else { Console.WriteLine("no"); }

}


static void testGetAllCountries() {

DataTable dataTable=clsCountry.GetAllCountries();

foreach (DataRow row in dataTable.Rows) {
Console.WriteLine($"{row["CountryID"]} ,{row["CountryName"]}");
}


}


static void Main(string[] args)
{
// testFindContact(1);
//  testAddNewContact();
//testUpdateContact(1);
//testDeleteContact(100);
```

```
//ListContacts();
//isContactExist(1);

//testFindCountryByID(1);
//testFindCountryByName("Egypt");

testAddNewCountry();
//testUpdateCountry();
//testDeleteCountry(1007);
//testGetAllCountries();
//isCountryExistByID(1);
//isCountryExistByName("Egypt");
Console.ReadKey();
}
}
}
```

Homework 2: Add fields to the Country

Homework 2:

Add the following fields to the countries table in the database:

1- Code nvarchar(3) allow null

2- PhoneCode nvarchar(3) allow null

then Update the business and data access accordingly.

Homwork 2: Solution

ده كود ال sql عشان تعدل عالجدول

```
alter table Countries
add Code nvarchar(3) null,
PhoneCode nvarchar(3) null
```

ده كود ال DataAccess

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace ContactsDataAccessLayer
{
public class clsCountryData
{
public static bool GetCountryInfoByID(int ID, ref  string CountryName,
ref string Code, ref string PhoneCode)
{
bool isFound = false;
```

```
SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = "SELECT * FROM Countries WHERE CountryID = @CountryID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@CountryID", ID);

try
{
connection.Open();
SqlDataReader reader = command.ExecuteReader();

if (reader.Read())
{

// The record was found
isFound = true;

CountryName = (string)reader["CountryName"];

if (reader["Code"] != DBNull.Value)
{
Code = (string)reader["Code"];
}
else
{
Code = "";
}

if (reader["PhoneCode"] != DBNull.Value)
{
PhoneCode = (string)reader["PhoneCode"];
}
else
{
PhoneCode = "";
}

}
else
{
// The record was not found
isFound = false;
}

reader.Close();


}
catch (Exception ex)
{
//Console.WriteLine("Error: " + ex.Message);
isFound = false;
}
finally
{
connection.Close();
}
```

```
return isFound;
}


public static bool GetCountryInfoByName(string CountryName, ref int ID,
ref string Code, ref string PhoneCode)
{
bool isFound = false;

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = "SELECT * FROM Countries WHERE CountryName = @CountryName";

SqlCommand command = new SqlCommand(query, connection);

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

try
{
connection.Open();
SqlDataReader reader = command.ExecuteReader();

if (reader.Read())
{

// The record was found
isFound = true;

ID = (int)reader["CountryID"];

if (reader["Code"] != DBNull.Value)
{
Code = (string)reader["Code"];
}
else
{
Code = "";
}

if (reader["PhoneCode"] != DBNull.Value)
{
PhoneCode = (string)reader["PhoneCode"];
}
else
{
PhoneCode = "";
}

}
else
{
// The record was not found
isFound = false;
}

reader.Close();



}
catch (Exception ex)
```

```
{
//Console.WriteLine("Error: " + ex.Message);
isFound = false;
}
finally
{
connection.Close();
}

return isFound;
}


public static int AddNewCountry(string CountryName,string Code, string PhoneCode)
{
//this function will return the new contact id if succeeded and -1 if not.
int CountryID = -1;

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = @"INSERT INTO Countries (CountryName,Code,PhoneCode)
VALUES (@CountryName,@Code,@PhoneCode);
SELECT SCOPE_IDENTITY();";

SqlCommand command = new SqlCommand(query, connection);

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

if (Code != "")
command.Parameters.AddWithValue("@Code", Code);
else
command.Parameters.AddWithValue("@Code", System.DBNull.Value);

if (PhoneCode != "")
command.Parameters.AddWithValue("@PhoneCode", PhoneCode);
else
command.Parameters.AddWithValue("@PhoneCode", System.DBNull.Value);


try
{
connection.Open();

object result = command.ExecuteScalar();


if (result != null && int.TryParse(result.ToString(), out int insertedID))
{
CountryID = insertedID;
}
}

catch (Exception ex)
{
//Console.WriteLine("Error: " + ex.Message);

}

finally
{
```

```
connection.Close();
}


return CountryID;
}

public static bool UpdateCountry(int ID,string CountryName,string Code,string PhoneCode)
{

int rowsAffected=0;
SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = @"Update  Countries
set CountryName=@CountryName,
Code=@Code,
PhoneCode=@PhoneCode
where CountryID = @CountryID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@CountryID", ID);
command.Parameters.AddWithValue("@CountryName", CountryName);
command.Parameters.AddWithValue("@Code", Code);
command.Parameters.AddWithValue("@PhoneCode", PhoneCode);

try
{
connection.Open();
rowsAffected = command.ExecuteNonQuery();

}
catch (Exception ex)
{
//Console.WriteLine("Error: " + ex.Message);
return false;
}

finally
{
connection.Close();
}

return (rowsAffected > 0);
}

public static DataTable GetAllCountries()
{

DataTable dt = new DataTable();
SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = "SELECT * FROM Countries order by CountryName";

SqlCommand command = new SqlCommand(query, connection);

try
{
connection.Open();
```

```
SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{
dt.Load(reader);
}

reader.Close();


}

catch (Exception ex)
{
// Console.WriteLine("Error: " + ex.Message);
}
finally
{
connection.Close();
}

return dt;

}

public  static bool DeleteCountry(int CountryID)
{

int rowsAffected=0;

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = @"Delete Countries
where CountryID = @CountryID";

SqlCommand command = new SqlCommand(query, connection);

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

try
{
connection.Open();

rowsAffected = command.ExecuteNonQuery();

}
catch (Exception ex)
{
// Console.WriteLine("Error: " + ex.Message);
}
finally
{

connection.Close();

}

return (rowsAffected > 0);
```

```
}

public static bool IsCountryExist(int ID)
{
bool isFound = false;

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = "SELECT Found=1 FROM Countries WHERE CountryID = @CountryID";

SqlCommand command = new SqlCommand(query, connection);

command.Parameters.AddWithValue("@CountryID", ID);

try
{
connection.Open();
SqlDataReader reader = command.ExecuteReader();

isFound = reader.HasRows;

reader.Close();
}
catch (Exception ex)
{
//Console.WriteLine("Error: " + ex.Message);
isFound = false;
}
finally
{
connection.Close();
}

return isFound;
}


public static bool IsCountryExist(string CountryName)
{
bool isFound = false;

SqlConnection connection = new SqlConnection(clsDataAccessSettings.ConnectionString);

string query = "SELECT Found=1 FROM Countries WHERE CountryName = @CountryName";

SqlCommand command = new SqlCommand(query, connection);

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

try
{
connection.Open();
SqlDataReader reader = command.ExecuteReader();

isFound = reader.HasRows;

reader.Close();
}
catch (Exception ex)
{
```

```
//Console.WriteLine("Error: " + ex.Message);
isFound = false;
}
finally
{
connection.Close();
}

return isFound;
}


}
}
```

ده كود ال business

```
using System;
using System.Data;
using ContactsDataAccessLayer;


namespace ContactsBusinessLayer
{
public class clsCountry
{

public enum enMode { AddNew = 0, Update = 1 };
public enMode Mode = enMode.AddNew;

public int ID { set; get; }
public string CountryName { set; get; }
public string Code { set; get; }
public string PhoneCode { set; get; }


public clsCountry()

{
this.ID = -1;
this.CountryName = "";

Mode = enMode.AddNew;

}

private clsCountry(int ID, string CountryName,string Code, string PhoneCode)

{
this.ID = ID;
this.CountryName = CountryName;
this.Code = Code;
this.PhoneCode = PhoneCode;

Mode = enMode.Update;

}

private bool _AddNewCountry()
{
//call DataAccess Layer

this.ID= clsCountryData.AddNewCountry(this.CountryName,this.Code,this.PhoneCode);

return (this.ID != -1);
```

```
}

private bool _UpdateContact()
{
//call DataAccess Layer

return clsCountryData.UpdateCountry(this.ID, this.CountryName,this.Code,this.PhoneCode);

}

public static clsCountry Find(int ID)
{

string CountryName="";
string Code = "";
string PhoneCode = "";


int CountryID=-1;

if (clsCountryData.GetCountryInfoByID(ID,ref CountryName, ref Code, ref PhoneCode))

return new clsCountry(ID, CountryName,Code,PhoneCode);
else
return null;

}

public static clsCountry Find(string CountryName)
{

int ID = -1;
string Code = "";
string PhoneCode = "";


if (clsCountryData.GetCountryInfoByName(CountryName, ref ID,ref Code, ref PhoneCode))

return new clsCountry(ID, CountryName, Code, PhoneCode);
else
return null;

}


public bool Save()
{


switch  (Mode)
{
case enMode.AddNew:
if (_AddNewCountry())
{

Mode = enMode.Update;
return true;
}
else
{
return false;
}

case enMode.Update:

return _UpdateContact();

}
```

```
return false;
}

public static DataTable GetAllCountries()
{
return clsCountryData.GetAllCountries();

}

public static bool DeleteCountry(int ID)
{
return  clsCountryData.DeleteCountry(ID);
}

public static bool isCountryExist(int ID)
{
return clsCountryData.IsCountryExist(ID);
}

public static bool isCountryExist(string CountryName)
{
return clsCountryData.IsCountryExist(CountryName);
}



}
}
```

ده ال presentation

```
using System;
using System.Data;
using ContactsBusinessLayer;

namespace ContactsConsolApp
{
internal class Program
{
static void testFindContact(int ID)

{
clsContact Contact1 = clsContact.Find(ID);

if (Contact1 != null)
{
Console.WriteLine(Contact1.FirstName+ " " + Contact1.LastName);
Console.WriteLine(Contact1.Email);
Console.WriteLine(Contact1.Phone);
Console.WriteLine(Contact1.Address);
Console.WriteLine(Contact1.DateOfBirth);
Console.WriteLine(Contact1.CountryID);
Console.WriteLine(Contact1.ImagePath);
}
else
{
Console.WriteLine("Contact [" + ID + "] Not found!");
}
}

static void testAddNewContact()


{
clsContact Contact1 = new clsContact();
```

```
Contact1.FirstName = "Fadi";
Contact1.LastName = "Maher";
Contact1.Email = "A@a.com";
Contact1.Phone = "010010";
Contact1.Address = "address1";
Contact1.DateOfBirth = new DateTime(1977, 11, 6, 10, 30, 0);
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save())
{

Console.WriteLine("Contact Added Successfully with id=" + Contact1.ID);
}

}

static void testUpdateContact(int ID)

{
clsContact Contact1 = clsContact.Find(ID);

if (Contact1 != null)
{
//update whatever info you want
Contact1.FirstName = "Lina";
Contact1.LastName = "Maher";
Contact1.Email = "A2@a.com";
Contact1.Phone = "2222";
Contact1.Address = "222";
Contact1.DateOfBirth = new DateTime(1977, 11, 6, 10, 30, 0);
Contact1.CountryID = 1;
Contact1.ImagePath = "";

if (Contact1.Save())
{

Console.WriteLine("Contact updated Successfully ");
}

}
else
{
Console.WriteLine("Not found!");
}
}

static void testDeleteContact(int ID)

{

if (clsContact.isContactExist(ID))

if (clsContact.DeleteContact(ID))

Console.WriteLine("Contact Deleted Successfully.");
else
Console.WriteLine("Faild to delete contact.");

else
Console.WriteLine("The contact with id = " + ID + " is not found");

}

static void ListContacts()
{

DataTable dataTable = clsContact.GetAllContacts();
```

```
Console.WriteLine("Contacts Data:");

foreach (DataRow row in dataTable.Rows)
{
Console.WriteLine($"{row["ContactID"]},  {row["FirstName"]} {row["LastName"]}");
}

}

static void testIsContactExist(int ID)

{

if (clsContact.isContactExist(ID))

Console.WriteLine("Yes, Contact is there.");

else
Console.WriteLine("No, Contact Is not there.");

}

//---Test Country Business

static void testFindCountryByID(int ID)

{
clsCountry Country1 = clsCountry.Find(ID);

if (Country1 != null)
{
Console.WriteLine("Name: " + Country1.CountryName );
Console.WriteLine("Code: " + Country1.Code);
Console.WriteLine("PhoneCode: " + Country1.PhoneCode);

}

else
{
Console.WriteLine("Country [" + ID + "] Not found!");
}
}


static void testFindCountryByName(string CountryName)

{
clsCountry Country1 = clsCountry.Find(CountryName);

if (Country1 != null)
{
Console.WriteLine("Country [" + CountryName + "] isFound with ID = " +Country1.ID);
Console.WriteLine("Name: " + Country1.CountryName);
Console.WriteLine("Code: " + Country1.Code);
Console.WriteLine("PhoneCode: " + Country1.PhoneCode);
}

else
{
Console.WriteLine("Country [" + CountryName + "] Is Not found!");
}
}


static void testIsCountryExistByID(int ID)

{

if (clsCountry.isCountryExist(ID))
```

```
Console.WriteLine("Yes, Country is there.");

else
Console.WriteLine("No, Country Is not there.");

}

static void testIsCountryExistByName(string CountryName)

{

if (clsCountry.isCountryExist(CountryName))

Console.WriteLine("Yes, Country is there.");

else
Console.WriteLine("No, Country Is not there.");

}


static void testAddNewCountry()


{
clsCountry Country1 = new clsCountry();

Country1.CountryName = "Eygpt";
Country1.Code = "222";
Country1.PhoneCode = "001";


if (Country1.Save())
{

Console.WriteLine("Country Added Successfully with id=" + Country1.ID);
}

}

static void testUpdateCountry(int ID)

{
clsCountry Country1 = clsCountry.Find(ID);

if (Country1 != null)
{
//update whatever info you want
Country1.CountryName = "Egypt";
Country1.Code = "111";
Country1.PhoneCode = "555";


if (Country1.Save())
{

Console.WriteLine("Country updated Successfully ");
}

}
else
{
Console.WriteLine("Country is you want to update is Not found!");
}
}

static void testDeleteCountry(int ID)

{
```

```
if (clsCountry.isCountryExist(ID))

if (clsCountry.DeleteCountry(ID))

Console.WriteLine("Country Deleted Successfully.");
else
Console.WriteLine("Faild to delete Country.");

else
Console.WriteLine("Faild to delete: The Country with id = " + ID + " is not found");

}

static void ListCountries()
{

DataTable dataTable = clsCountry.GetAllCountries();

Console.WriteLine("Coutries Data:");

foreach (DataRow row in dataTable.Rows)
{
Console.WriteLine($"{row["CountryID"]},  {row["CountryName"]} , {row["Code"]}, {row["PhoneCode"]}");
}

}


static void Main(string[] args)
{

//   testFindContact(6);

// testAddNewContact();


//  testUpdateContact(1);

// testDeleteContact(100);

//  ListContacts();

//testIsContactExist(1);
// testIsContactExist(100);

//   testFindCountryByID(6);
//testFindCountryByID(100);
// testFindCountryByName("Egypt");
//testFindCountryByName("UK");

//testIsCountryExistByID(1);
//testIsCountryExistByID(100);

//testIsCountryExistByName("United States");
//testIsCountryExistByName("UK");

// testAddNewCountry();
//   testUpdateCountry(6);
// testDeleteCountry(6);
ListCountries();




Console.ReadKey();

}
}
}
```

Contacts - WinForms

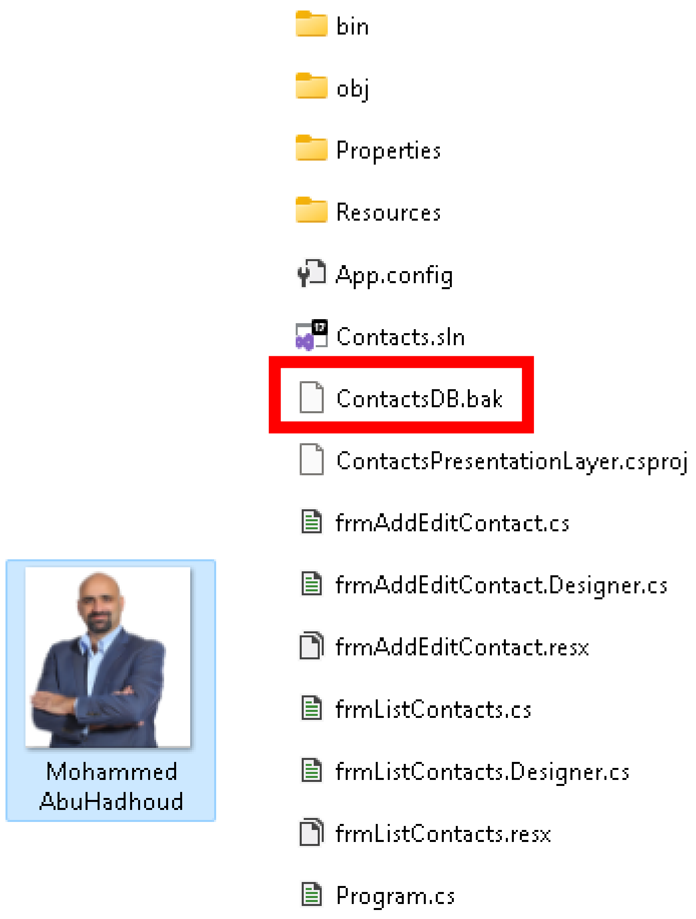
احنا دلوقتي عاملين ال data access layer وال business layer وعملنا عليهم console application

عاوزين بقي نعمل عليهم windows form

انا هستخدم ال data layer وال business layer والداتا بيز اللي عنده وهعمل عليهم

Windows form خطوه خطوه مع الكود بتاعه عشان افهم اكتر ايه اللي بيحصل

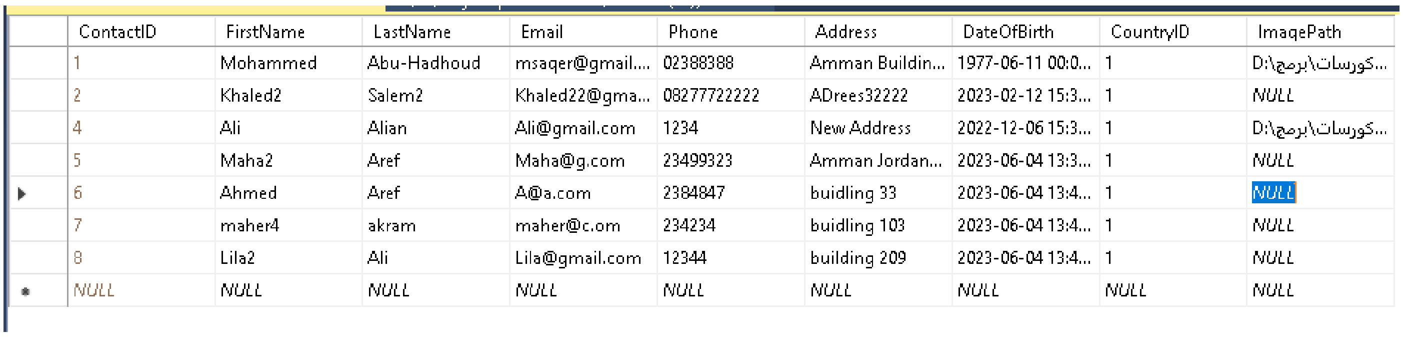
الداتابيز هوا حاططها هنا (استخدم اخر داتابيز كنت شغال عليها عشان الداتابيز دي مافيهاش اخر تعديل ياالما ضيف اخر عمودين اللي هما ال code وال phone code عشان ماتحصلش مشاكل)



bin
obj
Properties
Resources
App.config
Contacts.sln
ContactsDB.bak
ContactsPresentationLayer.csproj
frmAddEditContact.cs
frmAddEditContact.Designer.cs
frmAddEditContact.resx
frmListContacts.cs
frmListContacts.Designer.cs
frmListContacts.resx
Program.cs
Mohammed AbuHadhoud

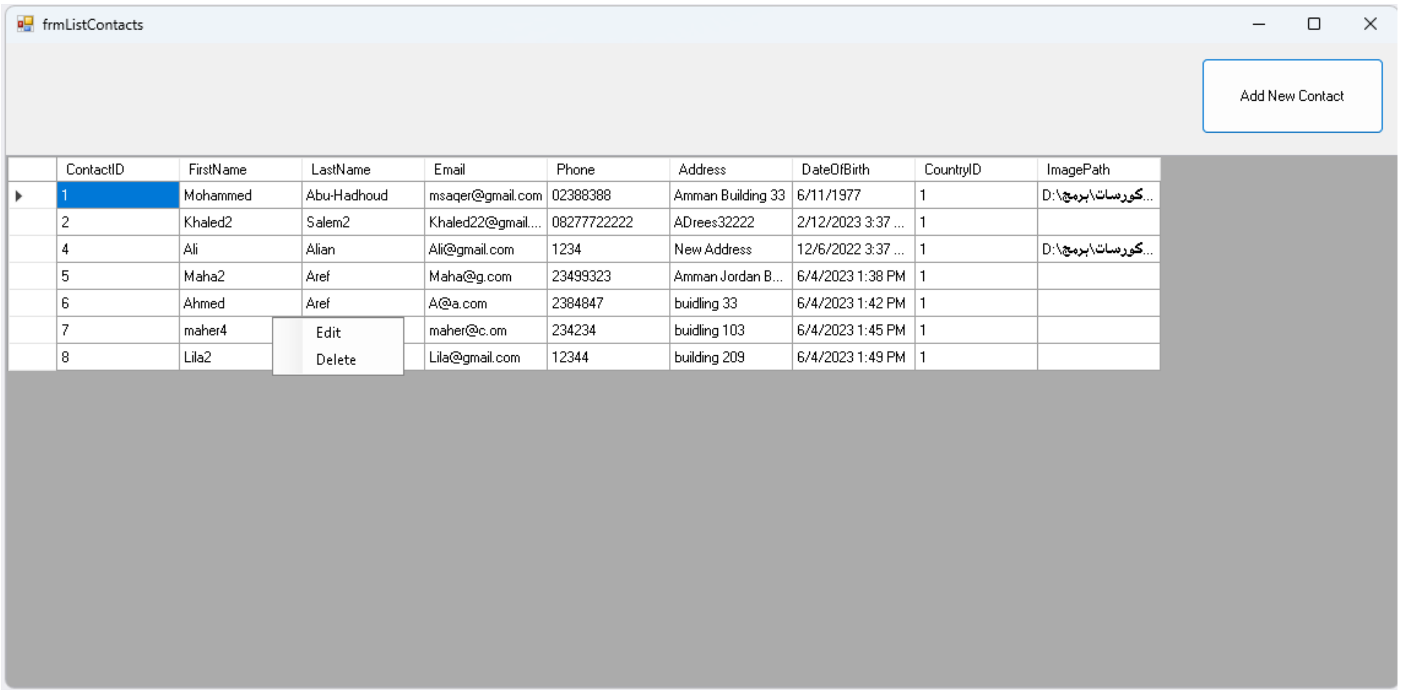
والصوره بتاعته موجوده في ملف ال resorces

وهروح للداتا بيز اعدل مسار الصورة في جدول ال contacts



	ContactID	FirstName	LastName	Email	Phone	Address	DateOfBirth	CountryID	ImagePath
	1	Mohammed	Abu-Hadhoud	msaqer@gmail....	02388388	Amman Buildin...	1977-06-11 00:0...	1	D:\برمج\كورسات...
	2	Khaled2	Salem2	Khaled22@gma...	08277722222	ADrees32222	2023-02-12 15:3...	1	NULL
	4	Ali	Alian	Ali@gmail.com	1234	New Address	2022-12-06 15:3...	1	D:\برمج\كورسات...
	5	Maha2	Aref	Maha@g.com	23499323	Amman Jordan...	2023-06-04 13:3...	1	NULL
▶	6	Ahmed	Aref	A@a.com	2384847	buidling 33	2023-06-04 13:4...	1	NULL
	7	maher4	akram	maher@c.om	234234	buidling 103	2023-06-04 13:4...	1	NULL
	8	Lila2	Ali	Lila@gmail.com	12344	building 209	2023-06-04 13:4...	1	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

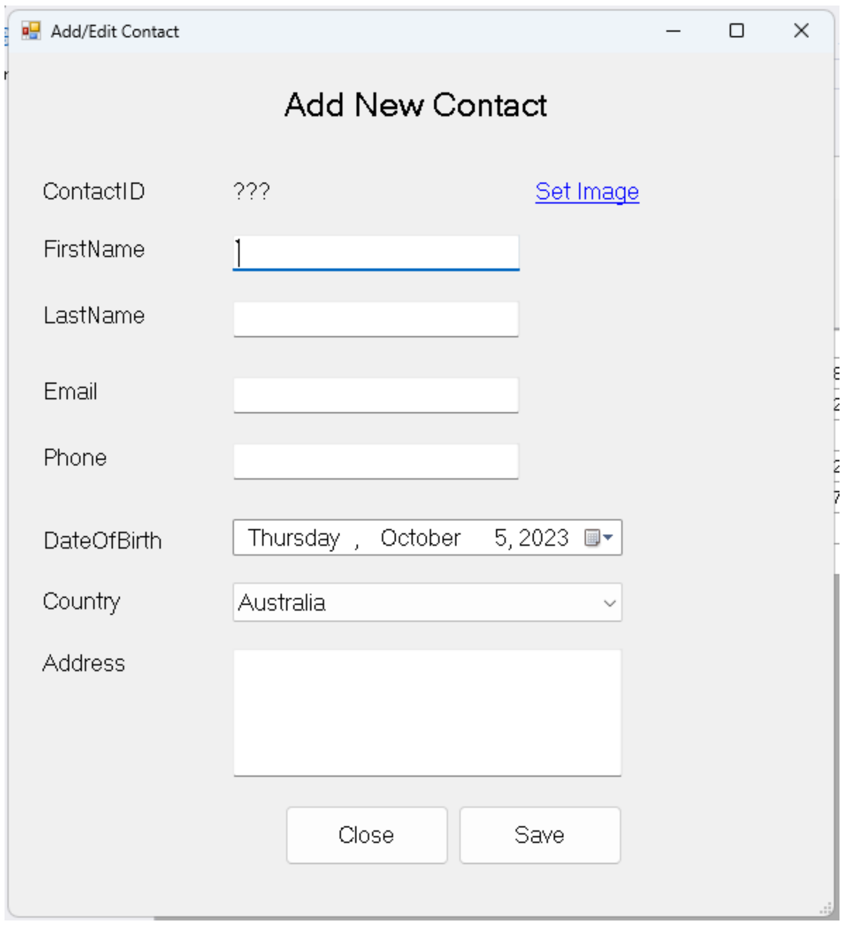
ده المشروع



frmListContacts

Add New Contact

	ContactID	FirstName	LastName	Email	Phone	Address	DateOfBirth	CountryID	ImagePath
▶	1	Mohammed	Abu-Hadhoud	msaqer@gmail.com	02388388	Amman Building 33	6/11/1977	1	D:\برمج\كورسات...
	2	Khaled2	Salem2	Khaled22@gmail....	08277722222	ADrees32222	2/12/2023 3:37 ...	1	
	4	Ali	Alian	Ali@gmail.com	1234	New Address	12/6/2022 3:37 ...	1	D:\برمج\كورسات...
	5	Maha2	Aref	Maha@g.com	23499323	Amman Jordan B...	6/4/2023 1:38 PM	1	
	6	Ahmed	Aref	A@a.com	2384847	buidling 33	6/4/2023 1:42 PM	1	
	7	maher4		maher@c.om	234234	buidling 103	6/4/2023 1:45 PM	1	
	8	Lila2		Lila@gmail.com	12344	building 209	6/4/2023 1:49 PM	1	



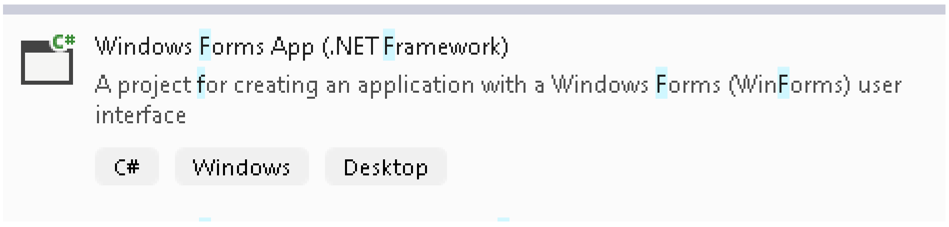
Add/Edit Contact

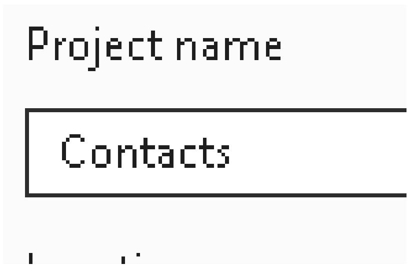
Add New Contact

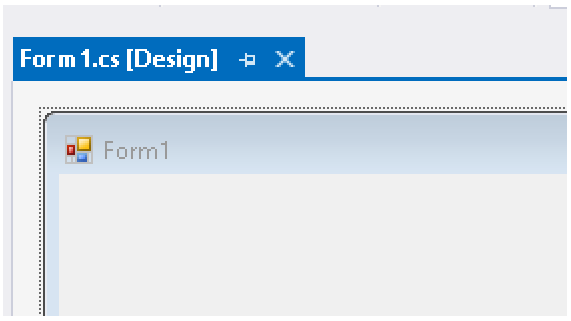
ContactID	???	Set Image
FirstName		
LastName		
Email		
Phone		
DateOfBirth	Thursday , October 5, 2023	
Country	Australia	
Address		

هوا طبعا فيه صورة بتظهر بس عشان محتاج اعمل reference عالمشروع بتاعه لاختلاف ال path بتاع المشاريع

طيب يلا بينا نعمل مشروع جديد ونسميه Contacts

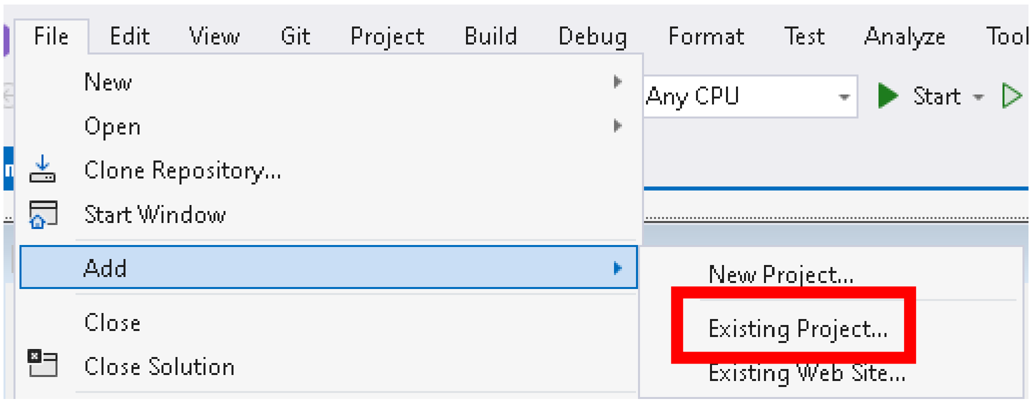

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C#
Windows
Desktop

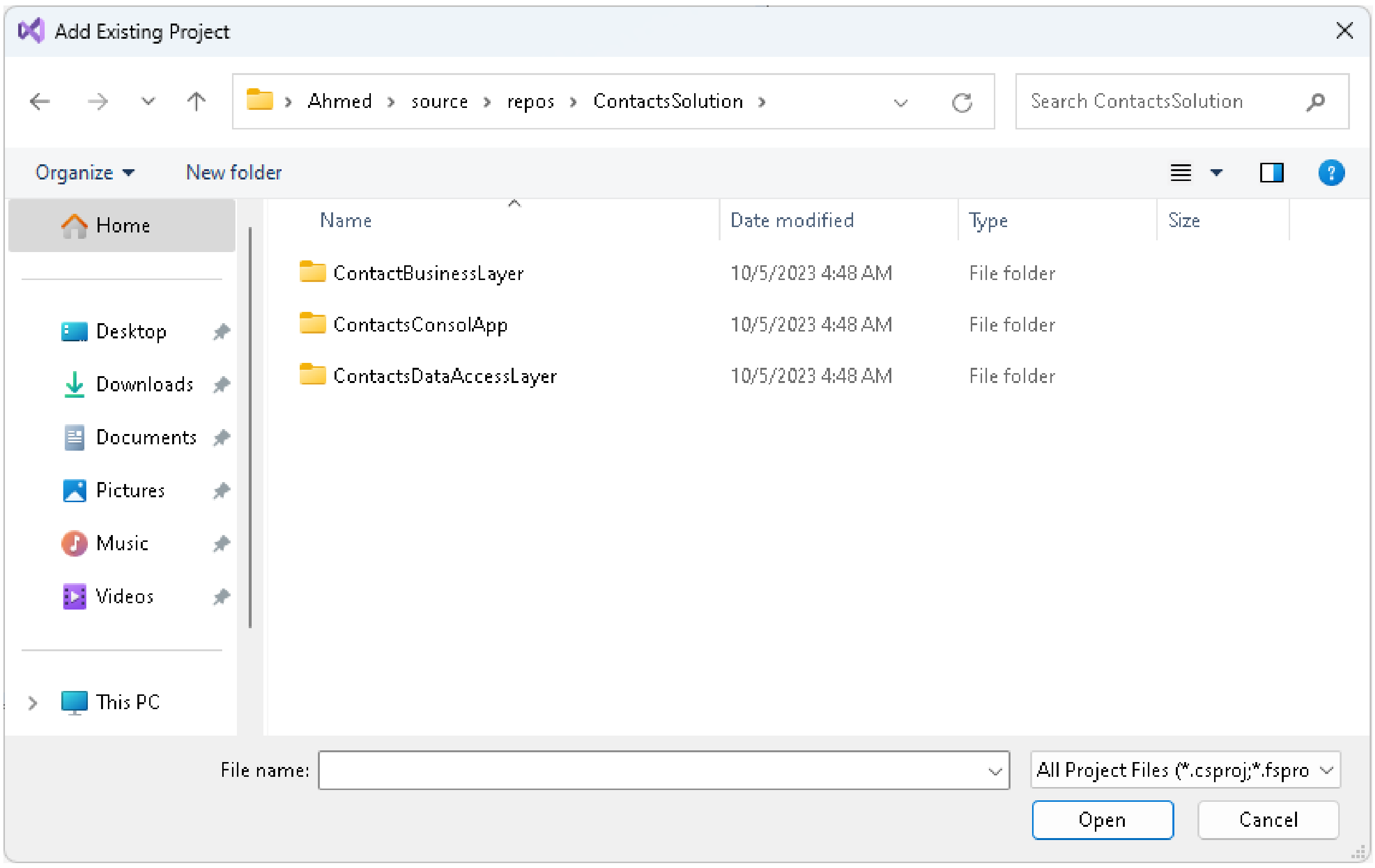




طيب قبل ما نعمل أي حاجه محتاجين نضيف الاتنين layer اللي عملناهم في ال console

هنضيفهم من file >> add>>existing project




Add Existing Project

Ahmed › source › repos › ContactsSolution ›

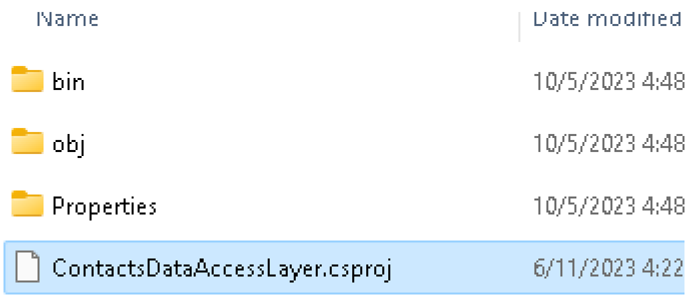
Search ContactsSolution

Organize ▾ New folder

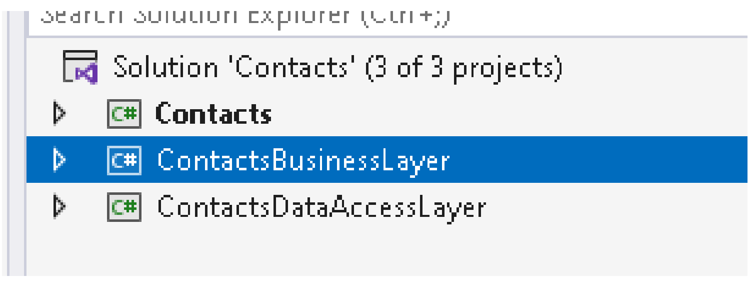
Name	Date modified	Type	Size
ContactBusinessLayer	10/5/2023 4:48 AM	File folder	
ContactsConsolApp	10/5/2023 4:48 AM	File folder	
ContactsDataAccessLayer	10/5/2023 4:48 AM	File folder	

File name:

All Project Files (*.csproj;*.fspro

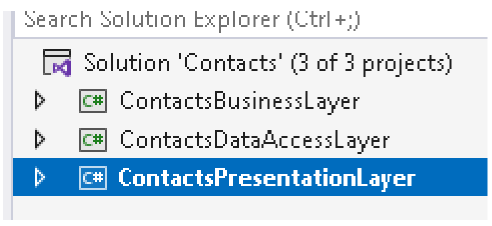


Name	Date modified
bin	10/5/2023 4:48
obj	10/5/2023 4:48
Properties	10/5/2023 4:48
ContactsDataAccessLayer.csproj	6/11/2023 4:22


Search Solution Explorer (Ctrl+;)

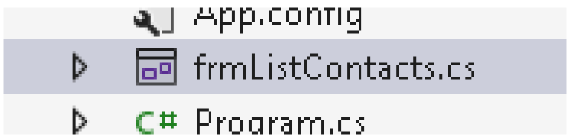
Solution 'Contacts' (3 of 3 projects)
- Contacts
- ContactsBusinessLayer
- ContactsDataAccessLayer

بعدين نغير اسم ال presentation layer من Contacts ل ContactsPresentationLayer


Search Solution Explorer (Ctrl+;)

Solution 'Contacts' (3 of 3 projects)
- ContactsBusinessLayer
- ContactsDataAccessLayer
- ContactsPresentationLayer

ونغير اسم الفورم


App.config
frmListContacts.cs
Program.cs

هنعمل ال reference لل layer بتاع ال presentation وبتاع ال business

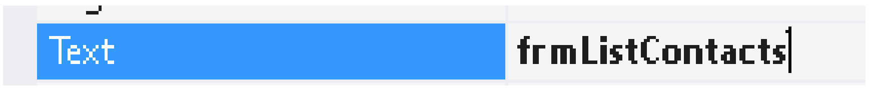
نيجي بقي للفورم

هنضيف فيه data grid view وده اللي هعرض فيه بيانات ال contacts وممكن تستعمل tree او list view عادي بس ده بتقدر تكتب فيه زي شيت الاكسيل كده

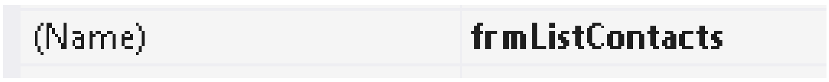
و هنضيف زرار من خلاله هنضيف contact جديد وهنضيف كمان context menu بحيث لما ادوس كليك يمين علي أي contact تطلعلي قايمه فيها تعديل او حذف

تعالي نحطهم مع بعض

دي اعدادات الفورم

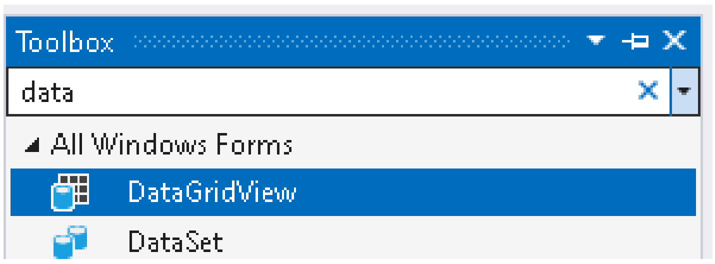


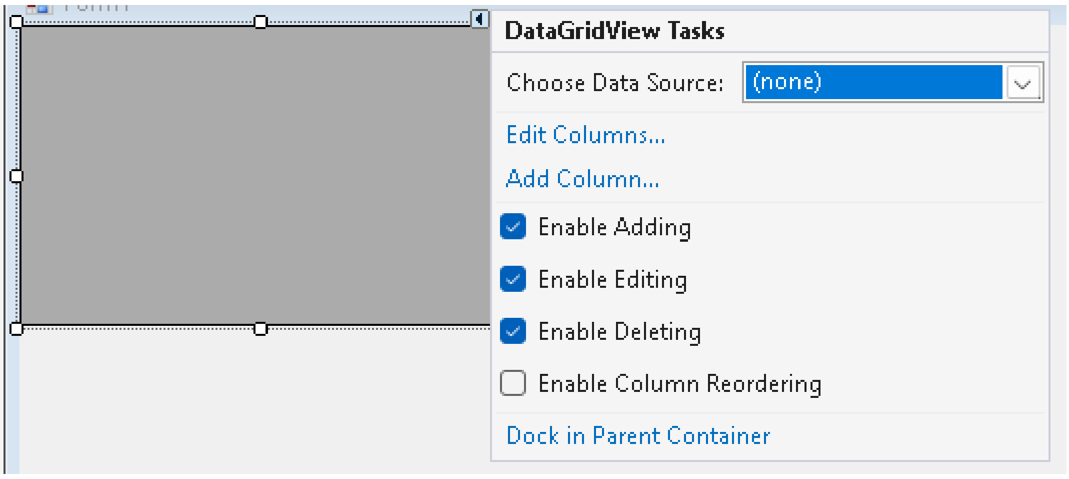
Text	frmListContacts



(Name)	frmListContacts

اول حاجه ال data grid

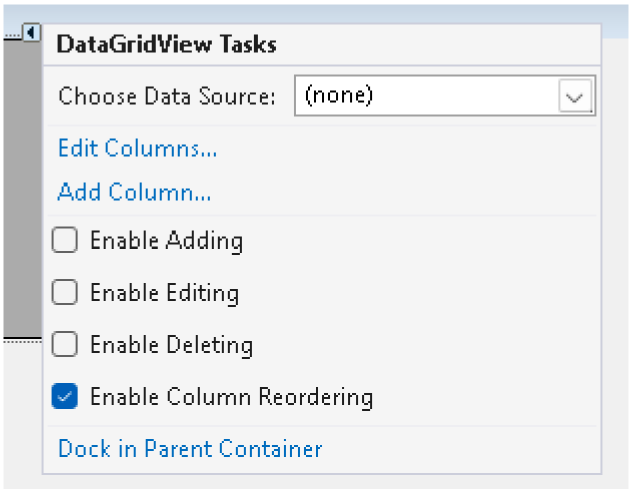



DataGridView Tasks

Choose Data Source: (none)

[✓] Enable Adding
[✓] Enable Editing
[✓] Enable Deleting
[ ] Enable Column Reordering

هنخلي الخيارات كده


DataGridView Tasks

Choose Data Source: (none)

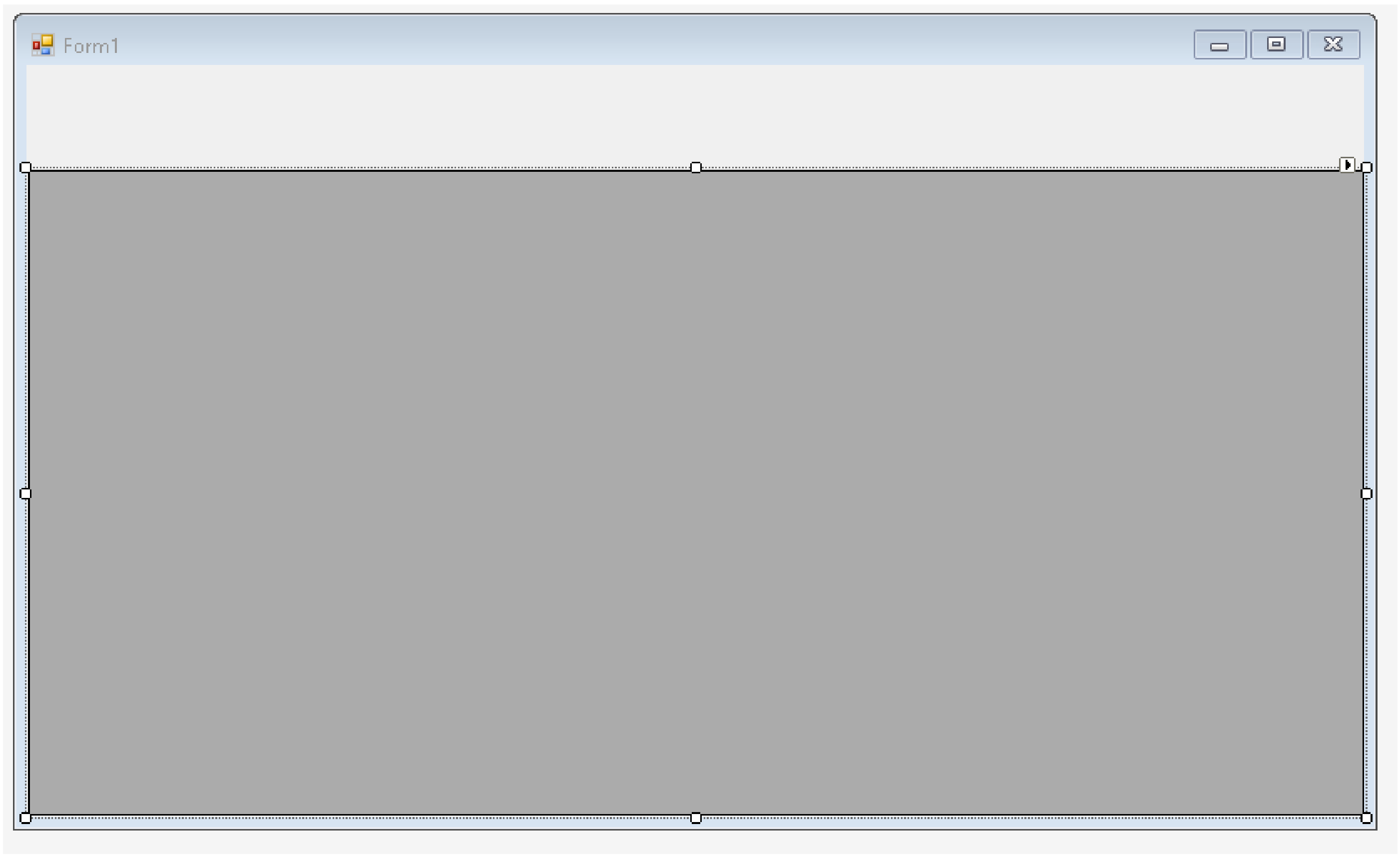
Edit Columns...

Add Column...

- [ ] Enable Adding
- [ ] Enable Editing
- [ ] Enable Deleting
- [x] Enable Column Reordering

Dock in Parent Container

عدلت الحجم والمكان



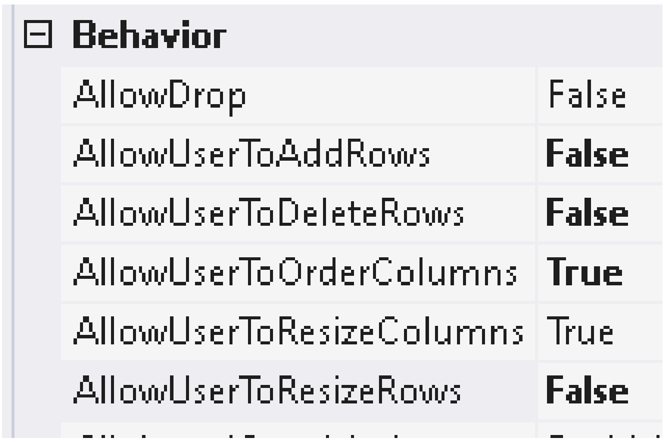
الخاصيه allow user to resize row دي بتخلي اليوزر يقدر يغير طول الصف

وال allow user to resize Columns بتخليه يقدر يغير عرض العمود

Allow drop دي بتخلي اليوزر يعمل drag and drop زي مثلا انه يحط صورة او ملف ماتشغلناش دلوقتي

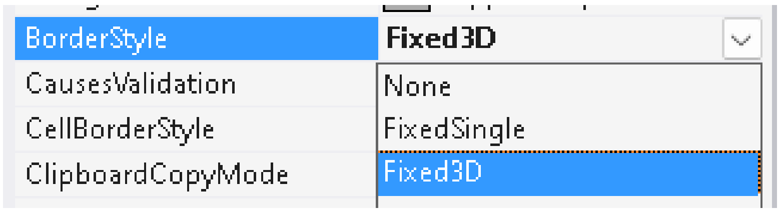
باقي الخصائص اللي في الصورة شارحه نفسها

هنعمل زي اللي في الصورة كده



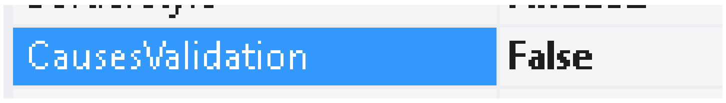
Behavior	
AllowDrop	False
AllowUserToAddRows	**False**
AllowUserToDeleteRows	**False**
AllowUserToOrderColumns	**True**
AllowUserToResizeColumns	True
AllowUserToResizeRows	**False**

ونعمل شكل ال border



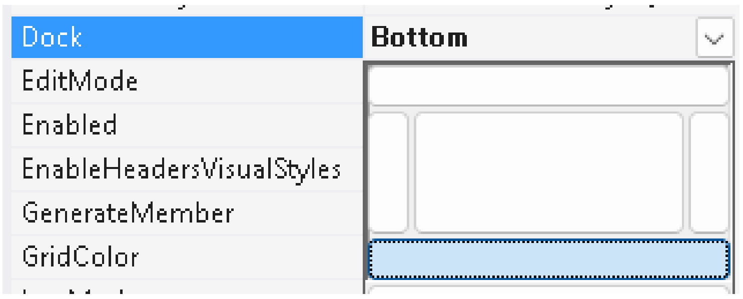
BorderStyle	**Fixed3D**
CausesValidation	
CellBorderStyle	
ClipboardCopyMode	

الخاصيه causes validation دي لواليوزر بيدخل داتا معينه وعايز تعملها validation طبعا احنا عاملينه يعرض داتا وبس فهنخليها false



CausesValidation	**False**

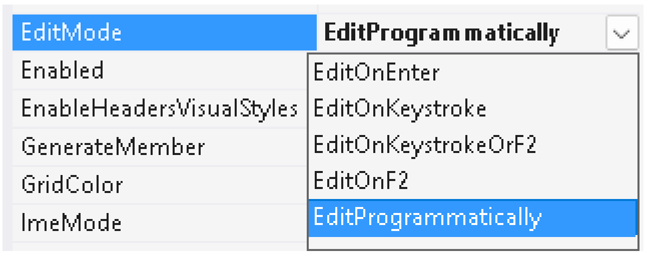
وبعدين ال dock ودي مكان ال view من الشاشه احنا عاملينها bottom



Dock	**Bottom**
EditMode	
Enabled	
EnableHeadersVisualStyles	
GenerateMember	
GridColor	

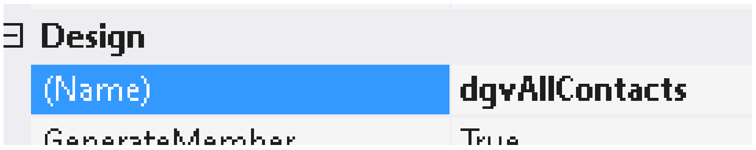
فيه ال edit mode وده بيقولك انت عايز عملية التعديل علي الداتا هتتم ازاي فاالحنا هنعملها

edit programmatically يعني هتتعدل برمجيا



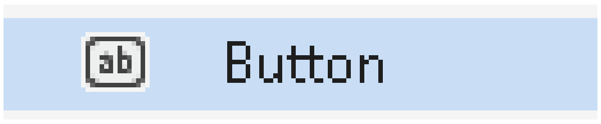
EditMode	**EditProgrammatically**
Enabled	
EnableHeadersVisualStyles	
GenerateMember	
GridColor	
ImeMode	

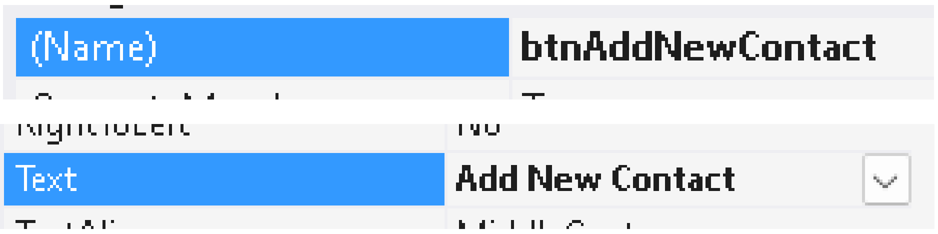
وبعدين نغير ال name



Design	
(Name)	**dgvAllContacts**
GenerateMember	True

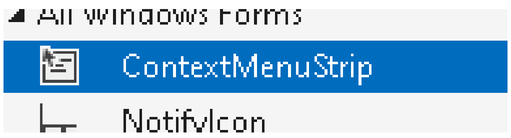
يلا نضيف الزرار

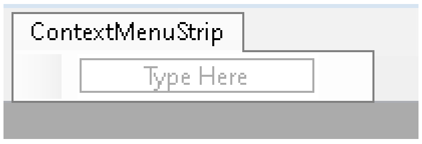


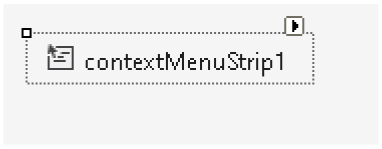

(Name) btnAddNewContact

Text Add New Contact

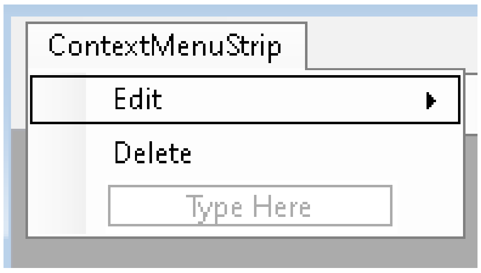
يلا نضيف ال context menu



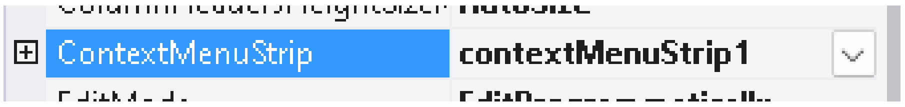




هتضيف العناصر بالماوس واحد edit والتاني delete

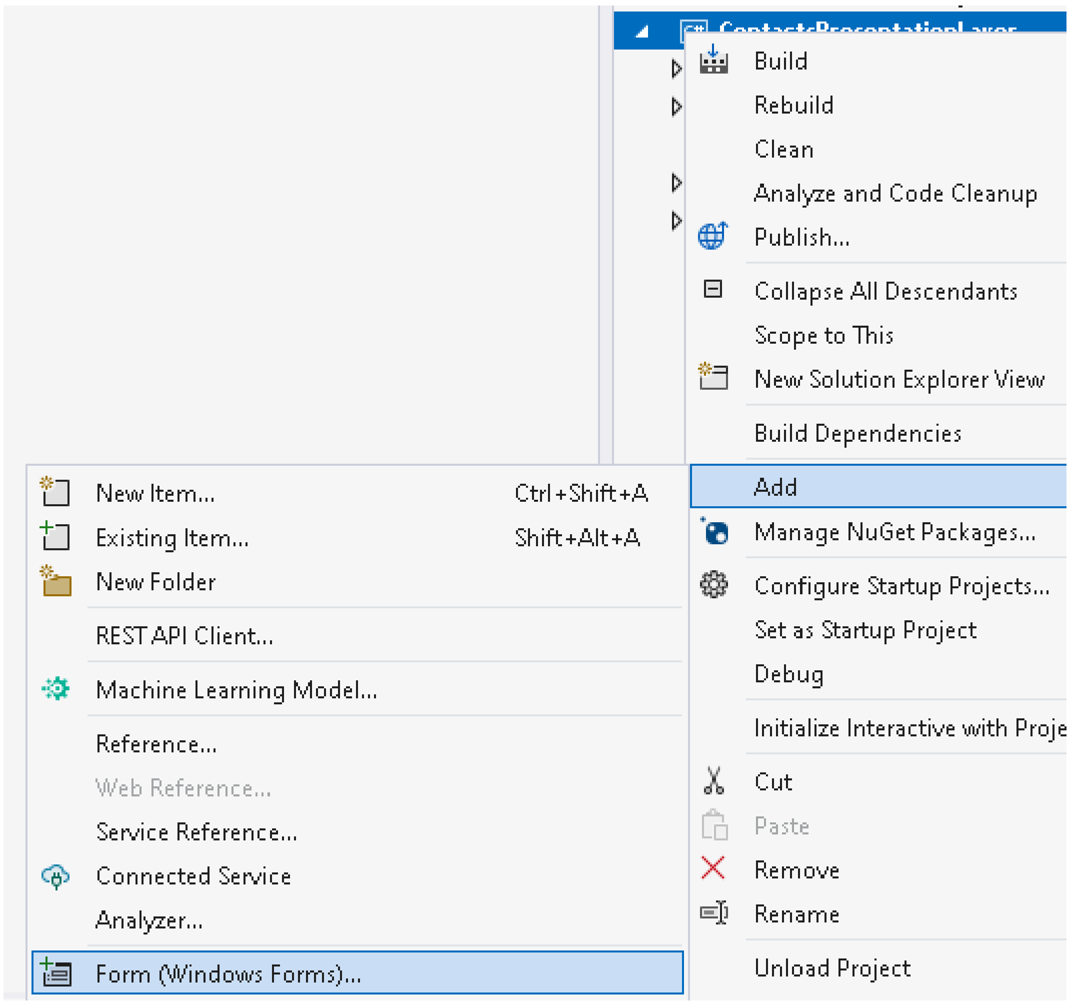


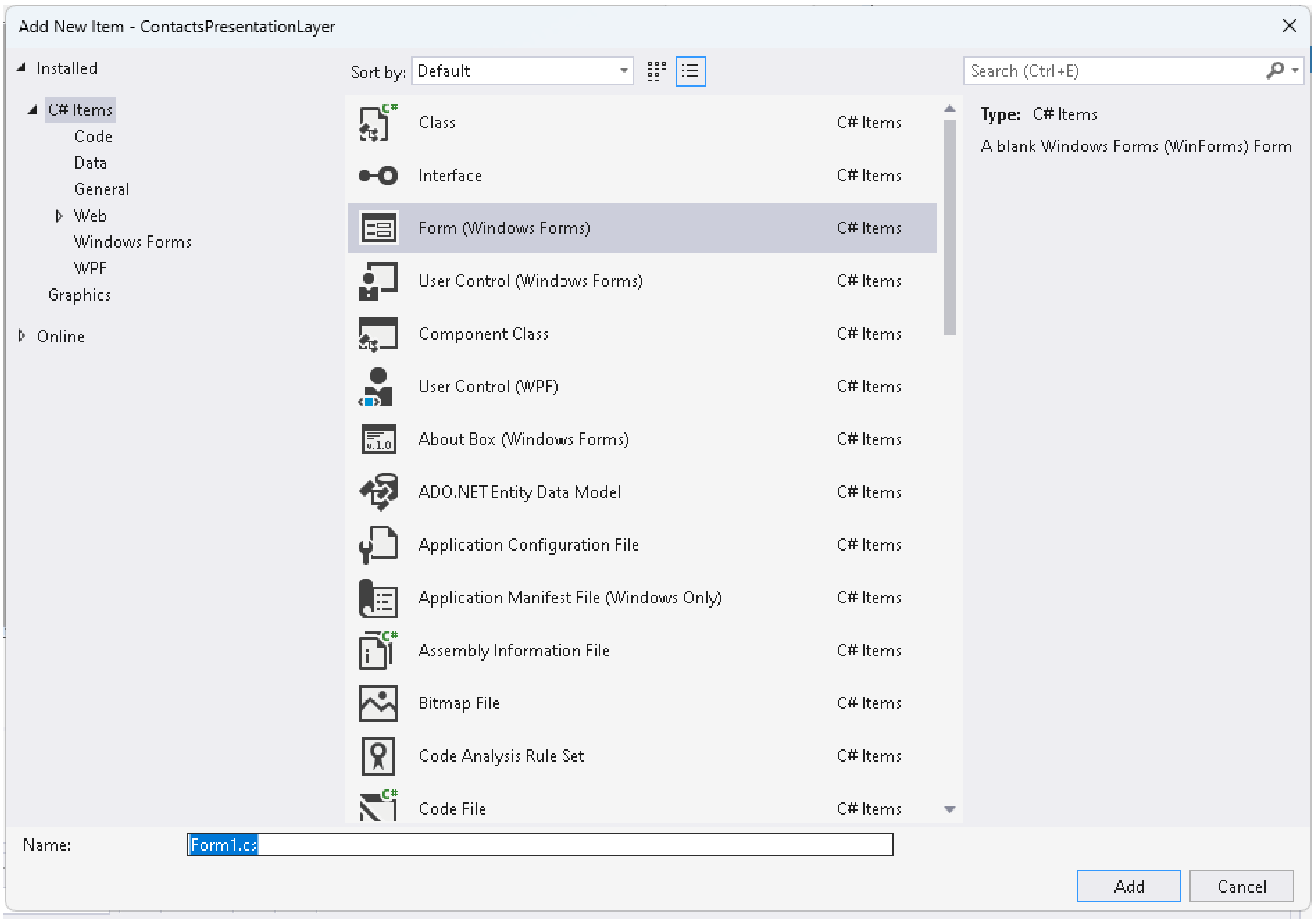
بعدين هنضيف ال menu لل grid view


ContextMenuStrip contextMenuStrip1

ده كده اول فورم

يلا نعمل الفورم التاني عشان نبقي خلصنا من الديزاين كله



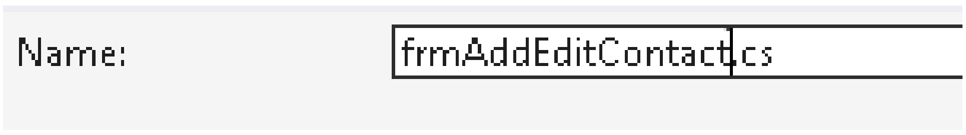

Add New Item - ContactsPresentationLayer
Installed
C# Items
Code
Data
General
Web
Windows Forms
WPF
Graphics
Online
Search (Ctrl+E)
Class C# Items
Interface C# Items
Form (Windows Forms) C# Items
User Control (Windows Forms) C# Items
Component Class C# Items
User Control (WPF) C# Items
About Box (Windows Forms) C# Items
ADO.NET Entity Data Model C# Items
Application Configuration File C# Items
Application Manifest File (Windows Only) C# Items
Assembly Information File C# Items
Bitmap File C# Items
Code Analysis Rule Set C# Items
Code File C# Items
Add
Cancel

Sort by: Default

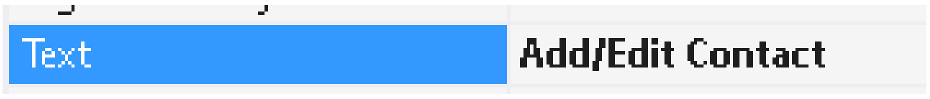
Type: C# Items

A blank Windows Forms (WinForms) Form

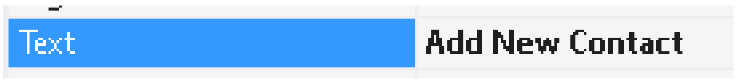
Name: Form1.cs

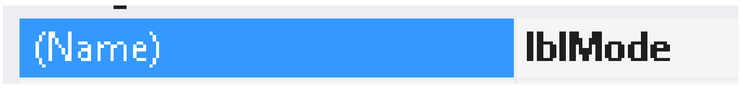

Name: frmAddEditContact.cs

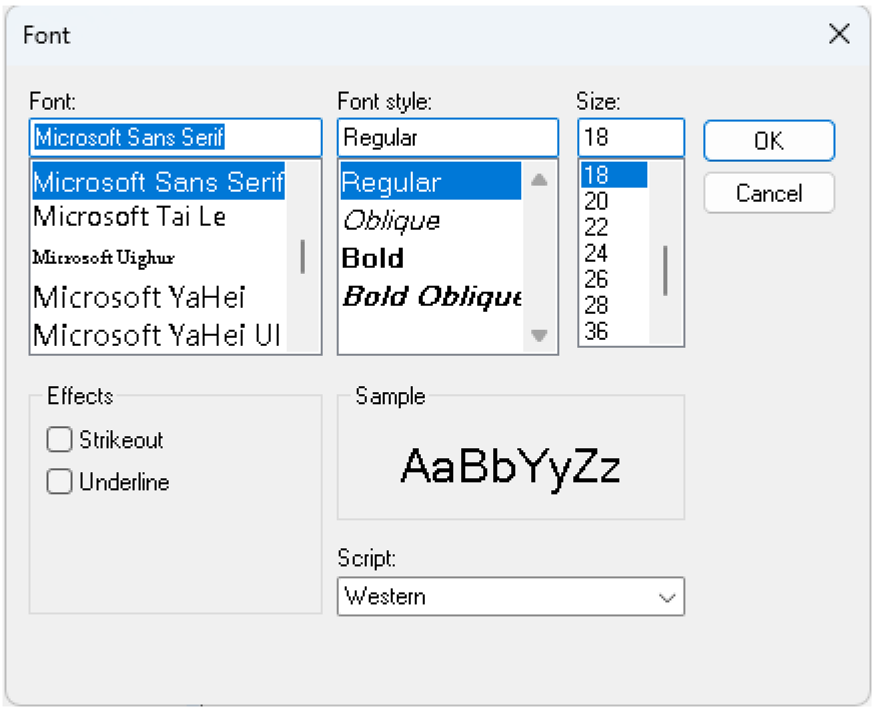
نيجي لاعدادات الفورم


Text | Add/Edit Contact

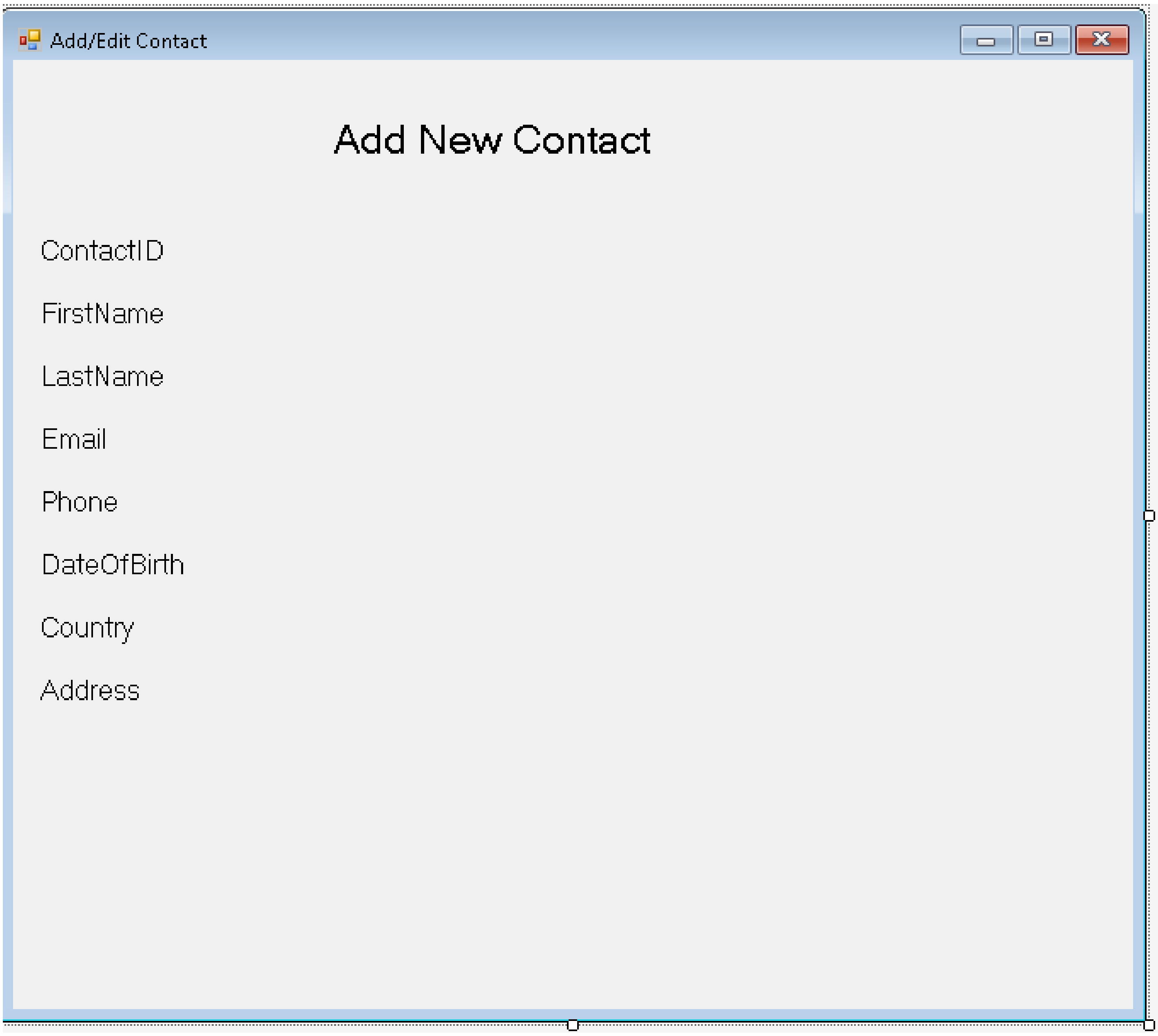
هنضيف label للعنوان


Text | Add New Contact


(Name) | lblMode


Font: Microsoft Sans Serif
Font style: Regular
Size: 18
Script: Western

بعدين هنضيف شوية labels هيبقوا عناوين وحجم الخط 12


Add/Edit Contact

Add New Contact

ContactID

FirstName

LastName

Email

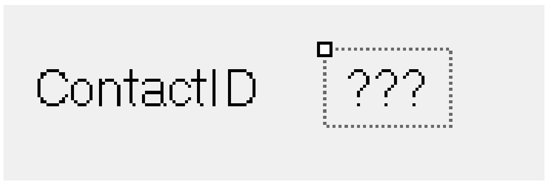
Phone

DateOfBirth

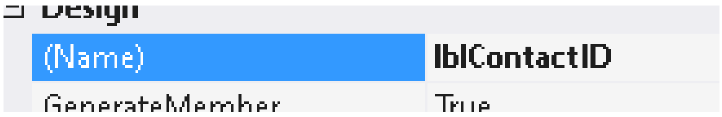
Country

Address

وهنضيف واحد تاني جنب ال contactID لان ده مش هتعدل فيه فخليه label احسن

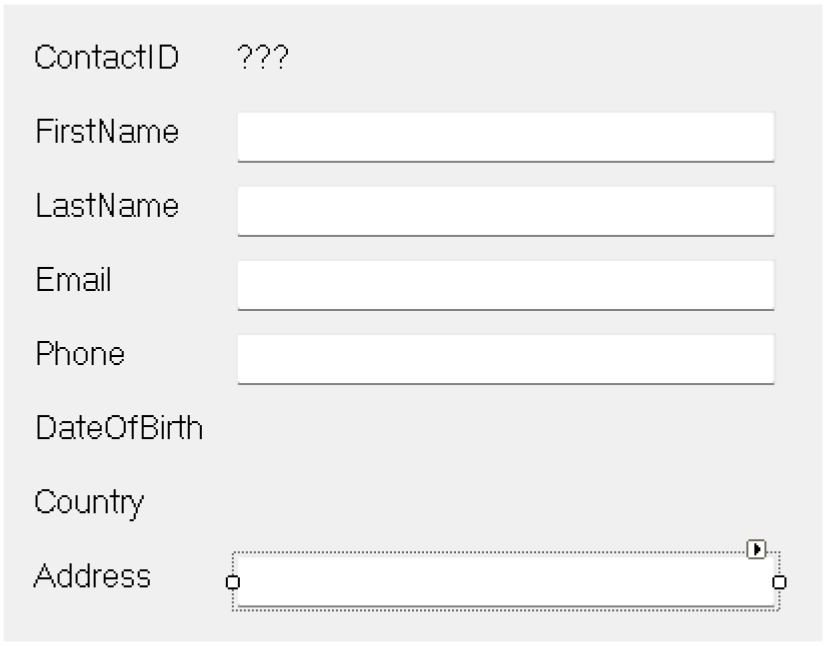

ContactID ???

ودي الخصائص بتاعته



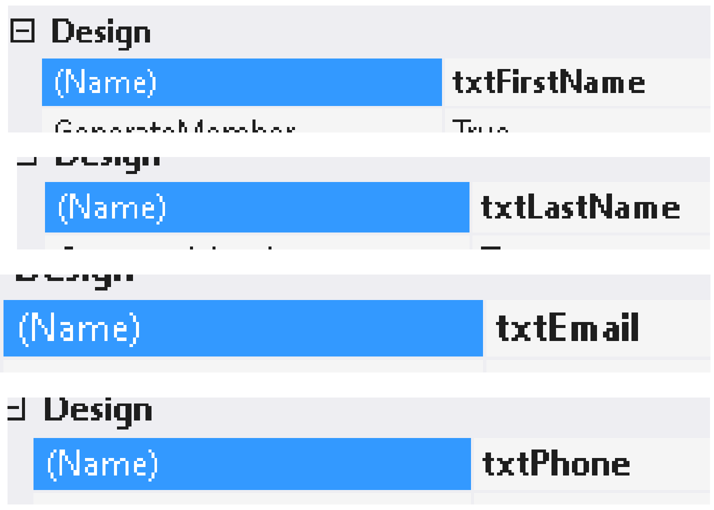
(Name)	lblContactID
GenerateMember	True

بعدين هنضيف text box لكل عنوان


ContactID ???
FirstName
LastName
Email
Phone
DateOfBirth
Country
Address

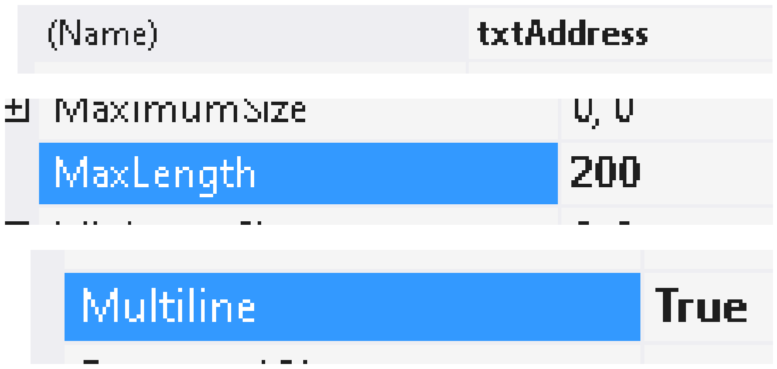
الخط بتاعهم حجمه 12

هنغير اسم كل واحد فيهم



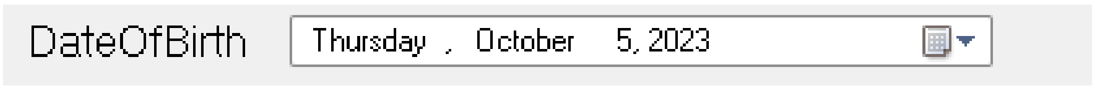
(Name)	txtFirstName
(Name)	txtLastName
(Name)	txtEmail
(Name)	txtPhone

ودي خصائص ال text box بتاع ال address

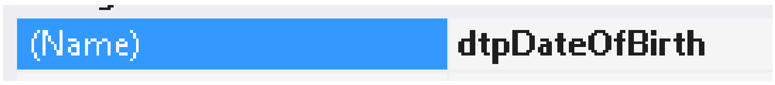


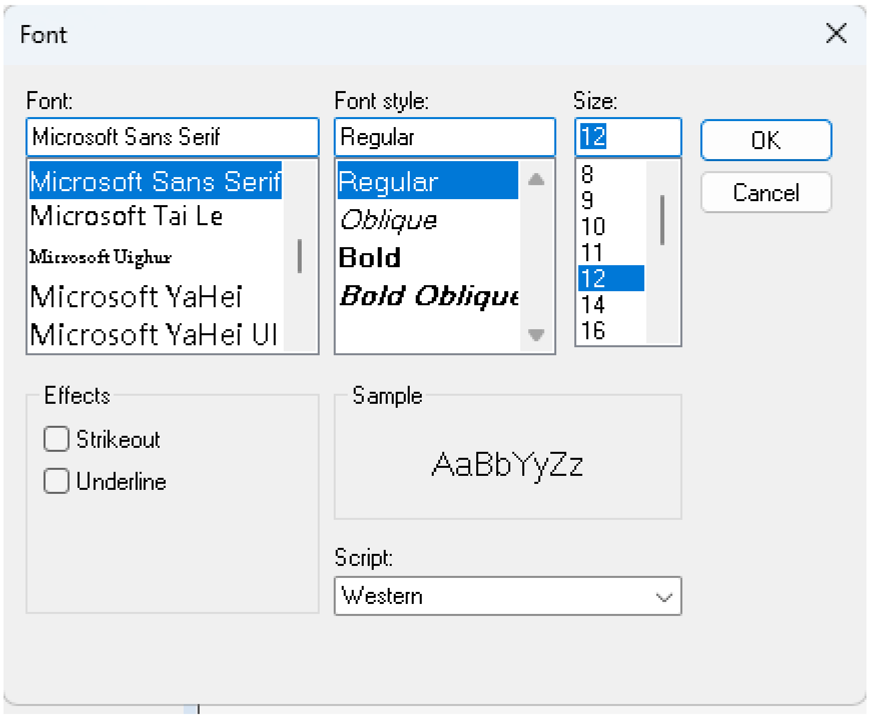
(Name)	txtAddress
MaximumSize	0, 0
MaxLength	200
Multiline	True

بالنسبه لل date of birth هنعمله dateTime Picker


DateOfBirth | Thursday , October 5, 2023

وdي الخصائص بتاعته


(Name) dtpDateOfBirth

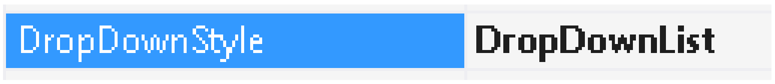

Font

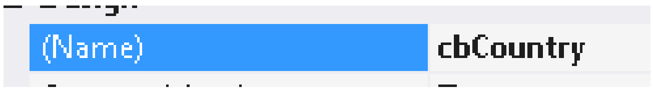
Font:	Font style:	Size:
Microsoft Sans Serif	Regular	12

Effects: Strikeout ☐, Underline ☐
Sample: AaBbYyZz
Script: Western

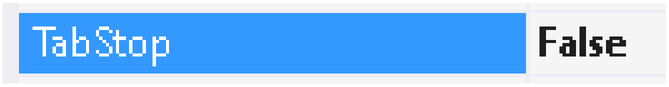
بالنسبه لل country دي هنعملها combo box

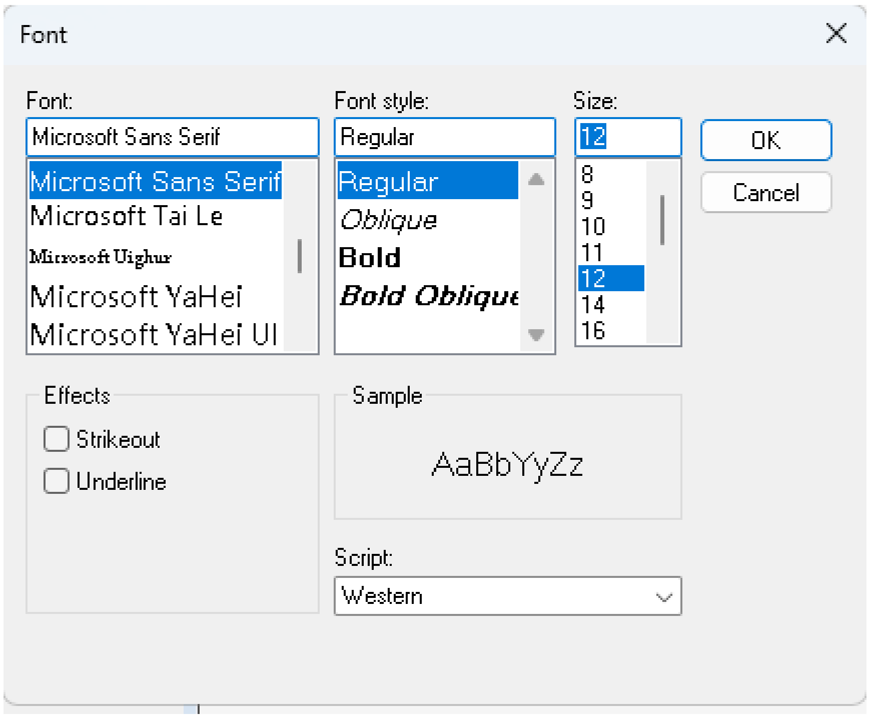
ودي الخصائص بتاعته


DropDownStyle DropDownList


(Name) cbCountry

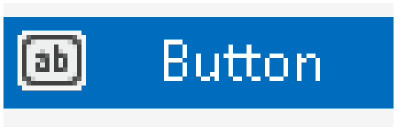
والخاصيه دي لما تعملها true اليوزر مش هيقدر يعمل عليه focus بزرار ال tab

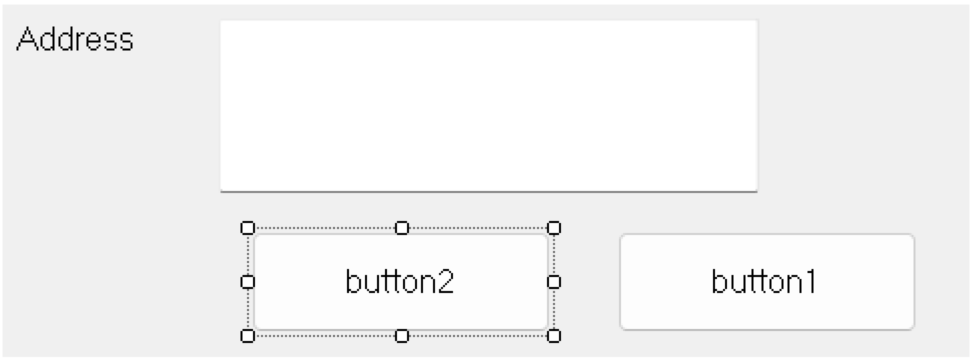

TabStop False

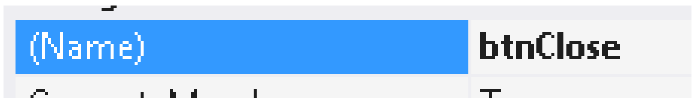

Font:
Microsoft Sans Serif
Font style:
Regular
Size:
12
Effects
Strikeout
Underline
Sample
AaBbYyZz
Script:
Western

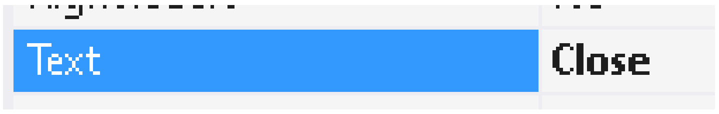
هنضيف زرارين واحد close والتاني save

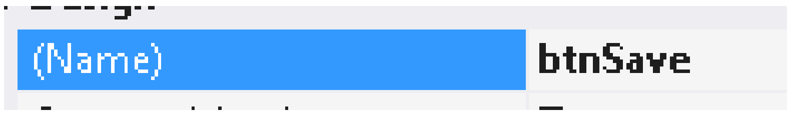
حجم الخط بتاعهم 12

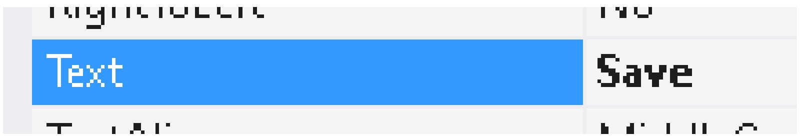




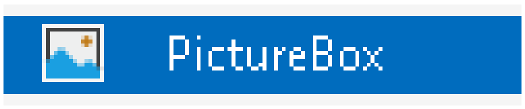

(Name) btnClose

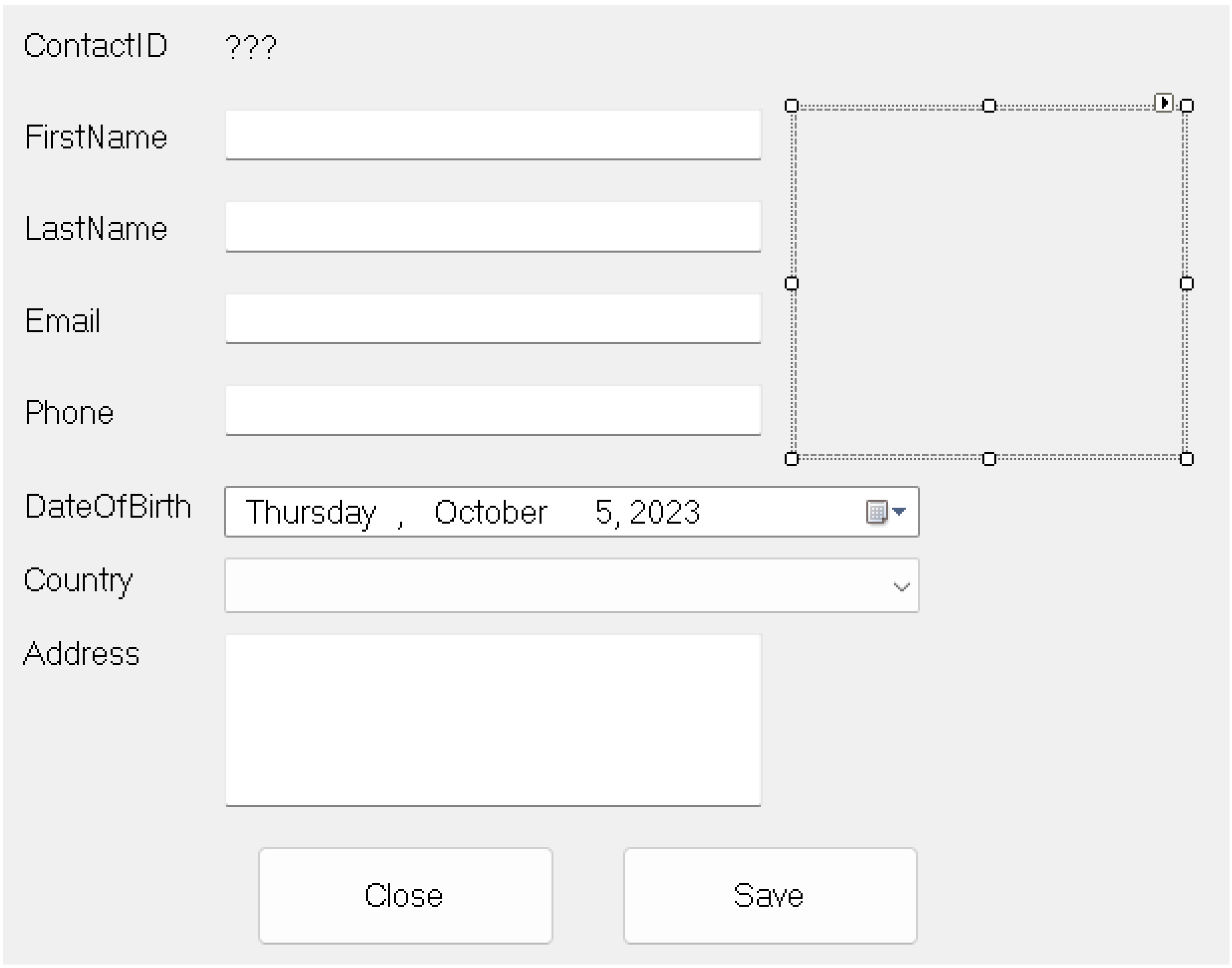

Text Close


(Name) btnSave

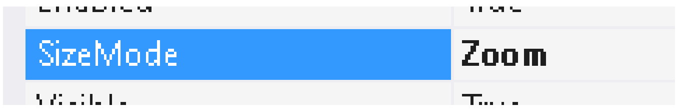

Text Save

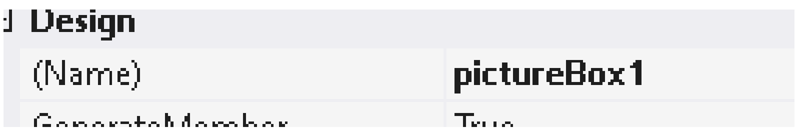
بعدين نضيف picture box



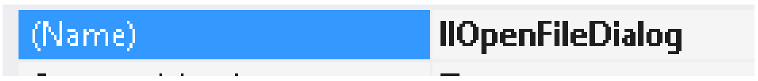


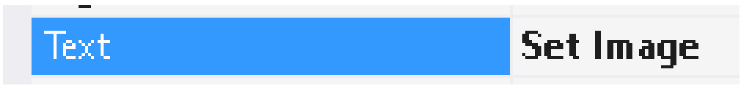
ودي الخصائص بتاعته

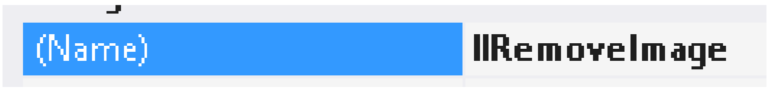

SizeMode Zoom

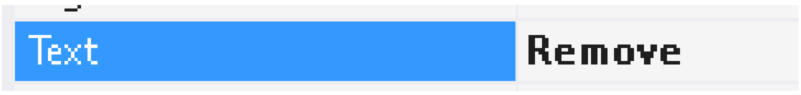

Design
(Name) pictureBox1

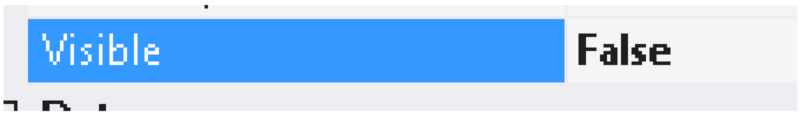
وفوق ال picture box هنضيف اتنين linked labl واحد يضيف صورة والتاني يشيلها وحجم الخط بتاعهم 12


(Name) llOpenFileDialog

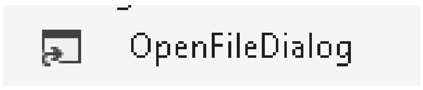

Text | Set Image


(Name) | llRemoveImage


Text | Remove

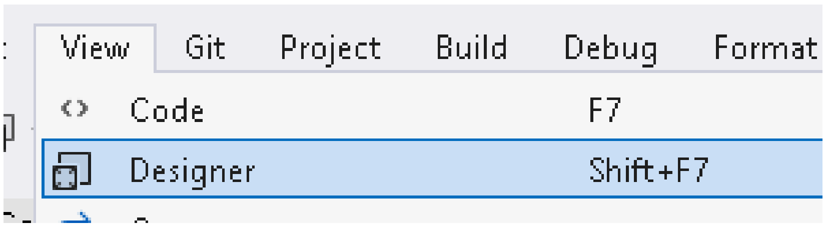

Visible | False

هنضيف file dialog

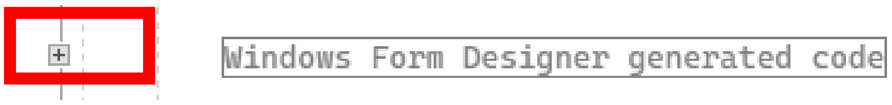


كده خلصنا الديزاين بتاع البرنامج

لو عايز تجيب قايمة بالخصائص بتاعت العناصر دي بتجيبها من قايمة view>> designer

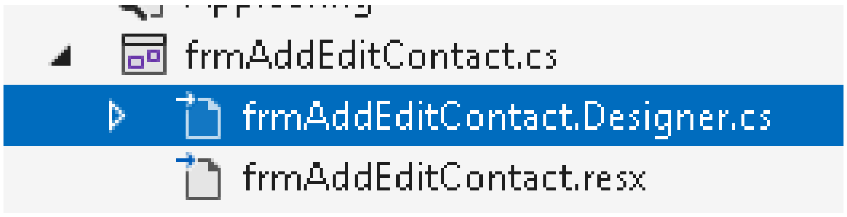


لما بيفتح الملف بتلاقي السطر ده بتدوس علي علامة ال + عشان تفتحه



بتلاقيهم جوه function اسمها InitializeComponent

او من طريقه تانيه و هيا من ال solution explorer



دي الصفحه اللي لما كنت بتحذف event يطلعلك خطأ تروح عندها تشيل السطر اللي تحتيه خط احمر

وده كود الخصائص بتاع الفورم الأساسي

```
private void InitializeComponent()
{
this.components = new System.ComponentModel.Container();
this.dgvAllContacts = new System.Windows.Forms.DataGridView();
this.btnAddNewContact = new System.Windows.Forms.Button();
this.contextMenuStrip1 = new System.Windows.Forms.ContextMenuStrip(this.components);
this.editToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.deleteToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
((System.ComponentModel.ISupportInitialize)(this.dgvAllContacts)).BeginInit();
this.contextMenuStrip1.SuspendLayout();
this.SuspendLayout();
//
// dgvAllContacts
//
this.dgvAllContacts.AllowUserToAddRows = false;
this.dgvAllContacts.AllowUserToDeleteRows = false;
this.dgvAllContacts.AllowUserToOrderColumns = true;
this.dgvAllContacts.AllowUserToResizeRows = false;
this.dgvAllContacts.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
this.dgvAllContacts.CausesValidation = false;
this.dgvAllContacts.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dgvAllContacts.ContextMenuStrip = this.contextMenuStrip1;
this.dgvAllContacts.Dock = System.Windows.Forms.DockStyle.Bottom;
this.dgvAllContacts.EditMode = System.Windows.Forms.DataGridViewEditMode.EditProgrammatically;
this.dgvAllContacts.Location = new System.Drawing.Point(0, 46);
this.dgvAllContacts.Name = "dgvAllContacts";
this.dgvAllContacts.ReadOnly = true;
this.dgvAllContacts.Size = new System.Drawing.Size(931, 434);
this.dgvAllContacts.TabIndex = 0;
//
// btnAddNewContact
//
this.btnAddNewContact.Location = new System.Drawing.Point(787, 12);
this.btnAddNewContact.Name = "btnAddNewContact";
this.btnAddNewContact.Size = new System.Drawing.Size(112, 23);
this.btnAddNewContact.TabIndex = 1;
this.btnAddNewContact.Text = "Add New Contact";
this.btnAddNewContact.UseVisualStyleBackColor = true;
//
// contextMenuStrip1
//
this.contextMenuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.editToolStripMenuItem,
this.deleteToolStripMenuItem});
this.contextMenuStrip1.Name = "contextMenuStrip1";
this.contextMenuStrip1.Size = new System.Drawing.Size(108, 48);
//
// editToolStripMenuItem
//
this.editToolStripMenuItem.Name = "editToolStripMenuItem";
this.editToolStripMenuItem.Size = new System.Drawing.Size(180, 22);
this.editToolStripMenuItem.Text = "Edit";
//
// deleteToolStripMenuItem
//
this.deleteToolStripMenuItem.Name = "deleteToolStripMenuItem";
this.deleteToolStripMenuItem.Size = new System.Drawing.Size(180, 22);
this.deleteToolStripMenuItem.Text = "Delete";
//
// frmListContacts
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(931, 480);
this.Controls.Add(this.btnAddNewContact);
this.Controls.Add(this.dgvAllContacts);
this.Name = "frmListContacts";
this.Text = "frmListContacts";
((System.ComponentModel.ISupportInitialize)(this.dgvAllContacts)).EndInit();
this.contextMenuStrip1.ResumeLayout(false);
this.ResumeLayout(false);

}
```

ودي الخصائص بتاعت الفورم بتاع التعديل والاضافه

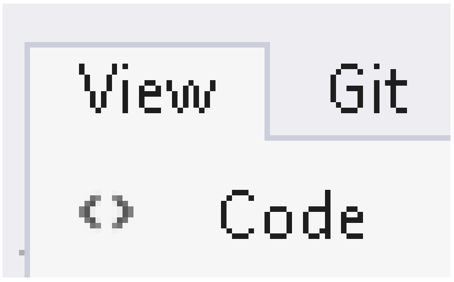
```
private void InitializeComponent()
{
this.lblMode = new System.Windows.Forms.Label();
this.label1 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.label4 = new System.Windows.Forms.Label();
this.label5 = new System.Windows.Forms.Label();
this.label6 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();
this.label8 = new System.Windows.Forms.Label();
this.lblContactID = new System.Windows.Forms.Label();
this.txtFirstName = new System.Windows.Forms.TextBox();
this.txtLastName = new System.Windows.Forms.TextBox();
this.txtPhone = new System.Windows.Forms.TextBox();
this.txtEmail = new System.Windows.Forms.TextBox();
this.txtAddress = new System.Windows.Forms.TextBox();
this.dtpDateOfBirth = new System.Windows.Forms.DateTimePicker();
this.cbCountry = new System.Windows.Forms.ComboBox();
this.btnSave = new System.Windows.Forms.Button();
this.btnClose = new System.Windows.Forms.Button();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.llOpenFileDialog = new System.Windows.Forms.LinkLabel();
this.llRemoveImage = new System.Windows.Forms.LinkLabel();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.SuspendLayout();
//
// lblMode
//
this.lblMode.AutoSize = true;
this.lblMode.Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lblMode.Location = new System.Drawing.Point(184, 32);
this.lblMode.Name = "lblMode";
this.lblMode.Size = new System.Drawing.Size(199, 29);
this.lblMode.TabIndex = 0;
this.lblMode.Text = "Add New Contact";
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label1.Location = new System.Drawing.Point(12, 97);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(82, 20);
this.label1.TabIndex = 1;
this.label1.Text = "ContactID";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label2.Location = new System.Drawing.Point(12, 143);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(82, 20);
this.label2.TabIndex = 2;
this.label2.Text = "FirstName";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label3.Location = new System.Drawing.Point(12, 235);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(48, 20);
this.label3.TabIndex = 4;
this.label3.Text = "Email";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label4.Location = new System.Drawing.Point(12, 189);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(82, 20);
this.label4.TabIndex = 3;
this.label4.Text = "LastName";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label5.Location = new System.Drawing.Point(12, 402);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(68, 20);
this.label5.TabIndex = 8;
this.label5.Text = "Address";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label6.Location = new System.Drawing.Point(12, 365);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(64, 20);
this.label6.TabIndex = 7;
this.label6.Text = "Country";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label7.Location = new System.Drawing.Point(12, 328);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(94, 20);
this.label7.TabIndex = 6;
this.label7.Text = "DateOfBirth";
//
// label8
//
this.label8.AutoSize = true;
this.label8.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.label8.Location = new System.Drawing.Point(12, 281);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(55, 20);
this.label8.TabIndex = 5;
this.label8.Text = "Phone";
//
// lblContactID
//
this.lblContactID.AutoSize = true;
this.lblContactID.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lblContactID.Location = new System.Drawing.Point(113, 98);
this.lblContactID.Name = "lblContactID";
this.lblContactID.Size = new System.Drawing.Size(36, 20);
this.lblContactID.TabIndex = 9;
this.lblContactID.Text = "???";
//
```

```
// txtFirstName
//
this.txtFirstName.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtFirstName.Location = new System.Drawing.Point(117, 139);
this.txtFirstName.Name = "txtFirstName";
this.txtFirstName.Size = new System.Drawing.Size(270, 26);
this.txtFirstName.TabIndex = 10;
//
// txtLastName
//
this.txtLastName.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtLastName.Location = new System.Drawing.Point(117, 185);
this.txtLastName.Name = "txtLastName";
this.txtLastName.Size = new System.Drawing.Size(270, 26);
this.txtLastName.TabIndex = 11;
//
// txtPhone
//
this.txtPhone.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtPhone.Location = new System.Drawing.Point(117, 277);
this.txtPhone.Name = "txtPhone";
this.txtPhone.Size = new System.Drawing.Size(270, 26);
this.txtPhone.TabIndex = 13;
//
// txtEmail
//
this.txtEmail.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtEmail.Location = new System.Drawing.Point(117, 231);
this.txtEmail.Name = "txtEmail";
this.txtEmail.Size = new System.Drawing.Size(270, 26);
this.txtEmail.TabIndex = 12;
//
// txtAddress
//
this.txtAddress.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txtAddress.Location = new System.Drawing.Point(117, 402);
this.txtAddress.MaxLength = 200;
this.txtAddress.Multiline = true;
this.txtAddress.Name = "txtAddress";
this.txtAddress.Size = new System.Drawing.Size(270, 87);
this.txtAddress.TabIndex = 15;
//
// dtpDateOfBirth
//
this.dtpDateOfBirth.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.dtpDateOfBirth.Location = new System.Drawing.Point(117, 328);
this.dtpDateOfBirth.Name = "dtpDateOfBirth";
this.dtpDateOfBirth.Size = new System.Drawing.Size(349, 26);
this.dtpDateOfBirth.TabIndex = 16;
//
// cbCountry
//
this.cbCountry.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cbCountry.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.cbCountry.FormattingEnabled = true;
this.cbCountry.Location = new System.Drawing.Point(117, 364);
this.cbCountry.Name = "cbCountry";
this.cbCountry.Size = new System.Drawing.Size(349, 28);
this.cbCountry.TabIndex = 17;
this.cbCountry.TabStop = false;
//
// btnSave
//
```

```
this.btnSave.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.btnSave.Location = new System.Drawing.Point(316, 508);
this.btnSave.Name = "btnSave";
this.btnSave.Size = new System.Drawing.Size(150, 51);
this.btnSave.TabIndex = 18;
this.btnSave.Text = "Save";
this.btnSave.UseVisualStyleBackColor = true;
//
// btnClose
//
this.btnClose.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.btnClose.Location = new System.Drawing.Point(133, 508);
this.btnClose.Name = "btnClose";
this.btnClose.Size = new System.Drawing.Size(150, 51);
this.btnClose.TabIndex = 19;
this.btnClose.Text = "Close";
this.btnClose.UseVisualStyleBackColor = true;
//
// pictureBox1
//
this.pictureBox1.Location = new System.Drawing.Point(403, 139);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(195, 174);
this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.Zoom;
this.pictureBox1.TabIndex = 20;
this.pictureBox1.TabStop = false;
//
// llOpenFileDialog
//
this.llOpenFileDialog.AutoSize = true;
this.llOpenFileDialog.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.llOpenFileDialog.Location = new System.Drawing.Point(400, 105);
this.llOpenFileDialog.Name = "llOpenFileDialog";
this.llOpenFileDialog.Size = new System.Drawing.Size(83, 20);
this.llOpenFileDialog.TabIndex = 21;
this.llOpenFileDialog.TabStop = true;
this.llOpenFileDialog.Text = "Set Image";
//
// llRemoveImage
//
this.llRemoveImage.AutoSize = true;
this.llRemoveImage.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.llRemoveImage.Location = new System.Drawing.Point(515, 105);
this.llRemoveImage.Name = "llRemoveImage";
this.llRemoveImage.Size = new System.Drawing.Size(68, 20);
this.llRemoveImage.TabIndex = 22;
this.llRemoveImage.TabStop = true;
this.llRemoveImage.Text = "Remove";
this.llRemoveImage.Visible = false;
//
// openFileDialog1
//
this.openFileDialog1.FileName = "openFileDialog1";
//
// frmAddEditContact
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(635, 610);
this.Controls.Add(this.llRemoveImage);
this.Controls.Add(this.llOpenFileDialog);
this.Controls.Add(this.pictureBox1);
this.Controls.Add(this.btnClose);
this.Controls.Add(this.btnSave);
this.Controls.Add(this.cbCountry);
this.Controls.Add(this.dtpDateOfBirth);
this.Controls.Add(this.txtAddress);
this.Controls.Add(this.txtPhone);
this.Controls.Add(this.txtEmail);
this.Controls.Add(this.txtLastName);
this.Controls.Add(this.txtFirstName);
this.Controls.Add(this.lblContactID);
this.Controls.Add(this.label5);
this.Controls.Add(this.label6);
this.Controls.Add(this.label7);
this.Controls.Add(this.label8);
this.Controls.Add(this.label3);
this.Controls.Add(this.label4);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.lblMode);
this.Name = "frmAddEditContact";
this.Text = "Add/Edit Contact";
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.ResumeLayout(false);
this.PerformLayout();

}
```

نروح بقي للكود

بنفتح صفحة الاكواد بتاعت الفورم من هنا



دي الاكواد بتاعت الفورم وهوا فاضي

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contacts
{
public partial class frmListContacts : Form
{
public frmListContacts()
{
InitializeComponent();
}
}
}
```

زي مانت شايف في ال constructor بيستدعي ال initialize components اللي كنا فاتحينها من ال designer والصفحه دي هيا عباره عن partial class يعني الكود بتاع الفورم مكتوب في اكتر من ملف والكلاس بيورث من كلاس تاني اسمه form

اول حاجه عاوزين نعملها اننا اول ماالفورم يفتح يروح يجيب الداتا من الداتا بيز عن طريق ال

getall contacts اللي عملناها في ال business layer

عشان كده هنعمل function تجيب الداتا وتعبيها في ال data grid

طيب هتعبيها ازاي؟

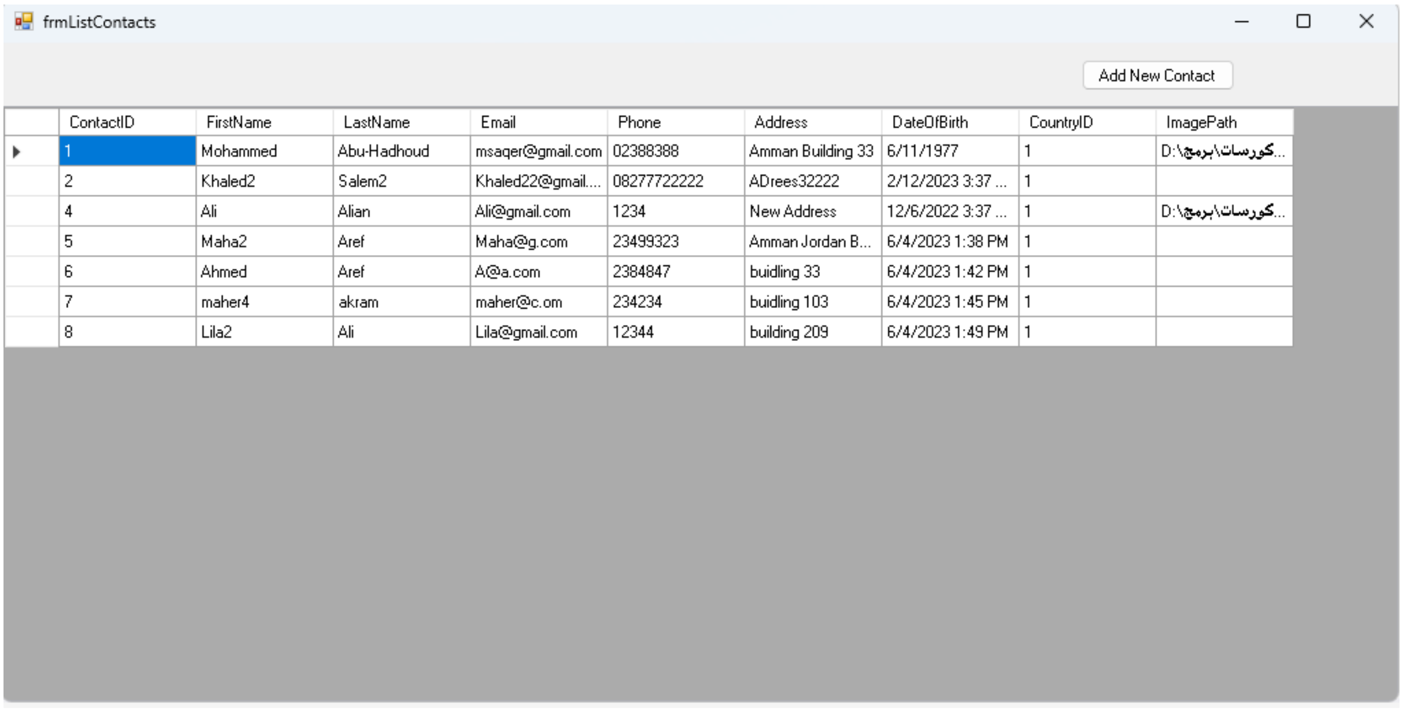
دي بقي اصعب شيء في الدنيا

ال data grid فيها object اسمه data source معموله get & set فانت كل اللي هتعمله انك هتستدعيه وتديله الداتا اللي خارجه كانك بتتعامل مع string

```
private void _RefreshContactsList() {
dgvAllContacts.DataSource=clsContact.GetAllContacts();
}
```

بعدين هنروح للخصائص بتاعت الفورم ونفتح event اسمه load ونستدعي فيه ال method ياإما عن طريق double click عالفورم نفسه

```
using ContactsBusinessLayer;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contacts
{
public partial class frmListContacts : Form
{
public frmListContacts()
{
InitializeComponent();
}

private void _RefreshContactsList() {
dgvAllContacts.DataSource=clsContact.GetAllContacts();
}

private void frmListContacts_Load(object sender, EventArgs e)
{
_RefreshContactsList();
}
}
}
```



	ContactID	FirstName	LastName	Email	Phone	Address	DateOfBirth	CountryID	ImagePath
▶	1	Mohammed	Abu-Hadhoud	msaqer@gmail.com	02388388	Amman Building 33	6/11/1977	1	D:\كورسات\برمج...
	2	Khaled2	Salem2	Khaled22@gmail....	08277722222	ADrees32222	2/12/2023 3:37 ...	1	
	4	Ali	Alian	Ali@gmail.com	1234	New Address	12/6/2022 3:37 ...	1	D:\كورسات\برمج...
	5	Maha2	Aref	Maha@g.com	23499323	Amman Jordan B...	6/4/2023 1:38 PM	1	
	6	Ahmed	Aref	A@a.com	2384847	buidling 33	6/4/2023 1:42 PM	1	
	7	maher4	akram	maher@c.om	234234	buidling 103	6/4/2023 1:45 PM	1	
	8	Lila2	Ali	Lila@gmail.com	12344	building 209	6/4/2023 1:49 PM	1	

تاني حاجه اني لما ادوس علي زرار ال add يفتح الفورم التانيولما اخلص اللي بعمله هنا وارجع للفورم الأساسي يعمل refresh للداتا

```
private void btnAddNewContact_Click(object sender, EventArgs e)
{
frmAddEditContact frm=new frmAddEditContact();
frm.ShowDialog();
_RefreshContactsList();
}
```

في ال edit بتاع ال context menu هعمل نفس الحاجه بس محتاج وانا رايح ابعتله ال id بتاع ال contact اللي انا واقف عليه عشان ياخده ويدور عليه ويجيبلي الداتا بتاعته

طيب هنعمل كده ازاي؟

كل اللي هتعمله انك هتروح علي الفورم بتاع الاضافه وتعدل عال constructor اللي فيه انه ياخد id ولما تيجي تستدعيه هيطلبه منك تديهوله

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contacts
{
public partial class frmAddEditContact : Form
{
public frmAddEditContact(int ID)
{
InitializeComponent();
}
}
}
```

طيب هجيب ال id اللي انا واقف عليه ازاي؟

فيه object اسمه current row ده موجود في ال data grid view ال current row ده موجود فيه حاجه اسمها cells هتستدعيه وتكتب رقم العمود اللي فيه ال id هوا عندنا رقمه صفر

زي كده

```
private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
int CurrentID = (int)dgvAllContacts.CurrentRow.Cells[0].Value;
frmAddEditContact frm = new frmAddEditContact(CurrentID);
frm.ShowDialog();
_RefreshContactsList();
}
```

هوا في الكود بتاعه مش عامل current id هوا حاطه في ال constructor علي طول بس انا عشان الكود يكون واضح

فيه كمان خطأ هيظهرلك في الكود بتاع زرار ال add new حط فيه سالب واحد

```
private void btnAddNewContact_Click(object sender, EventArgs e)
{
frmAddEditContact frm=new frmAddEditContact(-1);
frm.ShowDialog();
_RefreshContactsList();
}
```

طيب نيجي لل delete اللي موجوده ي ال context menu هنا احنا هنستدعي ال method بتاعت الحذف بس هنحط message box الأول عشان تاكد عاليوزر انك عاوز تحذف الرقم ده

```
private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{
if (MessageBox.Show("Are you sure you want to delete contact [" +
dgvAllContacts.CurrentRow.Cells[0].Value + "]","Confirm Delete",
MessageBoxButtons.OKCancel) == DialogResult.OK) {

if (clsContact.DeleteContact((int)dgvAllContacts.CurrentRow.Cells[0].Value))
{
MessageBox.Show("Contact Deleted Successfully.");
_RefreshContactsList();
}

else
{ MessageBox.Show("Contact is not deleted."); }

}
}
```

كده خلصنا من الفورم الأساسي وده الكود بتاعه كامل

```
using ContactsBusinessLayer;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contacts
{
public partial class frmListContacts : Form
{
public frmListContacts()
{
InitializeComponent();
}

private void _RefreshContactsList() {
dgvAllContacts.DataSource=clsContact.GetAllContacts();
}

private void frmListContacts_Load(object sender, EventArgs e)
{
_RefreshContactsList();
}

private void btnAddNewContact_Click(object sender, EventArgs e)
{
frmAddEditContact frm=new frmAddEditContact(-1);
frm.ShowDialog();
_RefreshContactsList();
}

private void editToolStripMenuItem_Click(object sender, EventArgs e)
{
int CurrentID = (int)dgvAllContacts.CurrentRow.Cells[0].Value;
frmAddEditContact frm = new frmAddEditContact(CurrentID);
frm.ShowDialog();
_RefreshContactsList();
}

private void deleteToolStripMenuItem_Click(object sender, EventArgs e)
{
if (MessageBox.Show("Are you sure you want to delete contact [" +
dgvAllContacts.CurrentRow.Cells[0].Value + "]","Confirm Delete",
MessageBoxButtons.OKCancel) == DialogResult.OK) {

if (clsContact.DeleteContact((int)dgvAllContacts.CurrentRow.Cells[0].Value))
{
MessageBox.Show("Contact Deleted Successfully.");
_RefreshContactsList();
}

else
{ MessageBox.Show("Contact is not deleted."); }

}
}
}
}
```

نيجي للفورم التاني

نبدأ بابسط حاجه وهيا زرار ال close

```
private void btnClose_Click(object sender, EventArgs e)
{
this.Close();
}
```

في الفورم ده انا عايز اعمل الاتي

1- اول مايفتح الفورم هيكون فيه رقم جايله عن طريق ال constructor لو الرقم ده -1 مش هيعمل حاجه ولو غير كده هيدور عالرقم ده في الداتا بيز ويجيب البيانات بتاعته
2- لو دوست علي زرار ال save محتاج اعمل addnew او update اللي هيفرق بين العمليتين حاجتين اول حاج هيا ال id اللي جاي عن طريق ال constructor تاني حاجه هيا ال method بتاع ال save نفسها اللي موجوده في ال business layer
3- لو مفيش صورة محطوطه هخفي ال label بتاع ال remove ولو فيه هظهرها
4- ال label بتاع ال set image بيفتح ال file explorer ولما اختار الصوره يخزن ال path ويعرضها
5- ال country تجيب كل البلدان اللي فيه جدول ال Countries

تعالي نجهز للكلام ده

اول حاجه هعمل enum لل mode عشان يسهل عليا عمليات الاضافه والتعديل وهعمل متغير استقبل فيه ال id اللي جايلي عن طريق ال constructor وهعمل object من الكلاس clsContact بس من غير new يعني يدوب هكتب اسمه وبعدين هملاه

```
enum enMode {AddNew=0,Update=1 };
enMode _Mode;

int _ContactID;
clsContact _Contact;
```

هعمل function تجيبلي كل الدول من جدول ال countries وتعبيها في القايمه بتاعت ال

combo box بتاع ال country بحيث يكون عندي قايمه بكل الدول اللي في الداتابيز اختار منهم

```
private void _FillCountriesInComboBox() {

DataTable dt= clsCountry.GetAllCountries();

foreach (DataRow row in dt.Rows) {

cbCountry.Items.Add(row["CountryName"];
}

}
```

هعمل function تدور عال id وتدور عليه في الداتابيز وتجيبلي بيانات ال contact وتعبيها في ال object اللي عملته وعاوزها بالمره تعبي ال combo box باسامي الدول وتختار الدوله اللي متخزنه في ال contact

طيب هجيب الدوله منين ؟

جدول ال contacts فيها عمود ال countryID هدور عليه في الداتابيز واعبيه

```
private void _LoadData() {

_FillCountriesInComboBox();
cbCountry.SelectedIndex = 0;

if (_Mode==enMode.AddNew) {
lblMode.Text = "Add New Contact";
_Contact=new clsContact();
return;
}

_Contact = clsContact.Find(_ContactID);

if (_Contact ==null) {

MessageBox.Show("This Form will be closed because no contact with ID = "+_ContactID);
this.Close();
return;
}

lblMode.Text = "Edit Contact ID = " + _ContactID;
lblContactID.Text=_ContactID.ToString();

txtFirstName.Text= _Contact.FirstName;
txtLastName.Text= _Contact.LastName;
txtEmail.Text= _Contact.Email;
txtPhone.Text= _Contact.Phone;
txtAddress.Text= _Contact.Address;
dtpDateOfBirth.Value = _Contact.DateOfBirth;

if (_Contact.ImagePath !="") {

pictureBox1.Load(_Contact.ImagePath);
}

llRemoveImage.Visible = (_Contact.ImagePath !="");
cbCountry.SelectedIndex = cbCountry.FindString( clsCountry.Find(_Contact.CountryID).CountryName);

}
```

كده جهزنا ال methods اللي هنشتغل بيها

هنروح دلوقتي لل constructor وعاوزين نعبي ال id ونعدل الmode

```
public frmAddEditContact(int ID)
{
InitializeComponent();

this._ContactID = ID;

if (ID==-1) {
this._Mode = enMode.AddNew;
} else {
this._Mode = enMode.Update;
}
}
```

عاوزين اول مالفورم يفتح يحمل الداتا

```
private void frmAddEditContact_Load(object sender, EventArgs e)
{
_LoadData();
}
```

عاوزين لما ندوس علي ال label بتاع ال set image يفتح ال file dialog ويرجعلي المسار بتاع الصوره

```
private void llOpenFileDialog_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
openFileDialog1.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.gif;*.bmp";
openFileDialog1.FilterIndex = 1;
openFileDialog1.RestoreDirectory = true;

if (openFileDialog1.ShowDialog()==DialogResult.OK) {

string selectedFilePath=openFileDialog1.FileName;
pictureBox1.Load(selectedFilePath);
}
}
```

عاوزين لما ندوس علي ال label بتاع remove يحذف الصوره ويختفي

طب ما كده مافيش تغيير حصل عالداتابيز ؟

ماحنا لما ييجي يدوس save هناخد من ال picture box المسار النهائي

```
private void llRemoveImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
pictureBox1.ImageLocation = null;
llRemoveImage.Visible = false;
}
```

عاوزين لما ندوس علي الزرار save يحفظ الداتا

```
private void btnSave_Click(object sender, EventArgs e)
{
int CountryID = clsCountry.Find(cbCountry.Text).ID;

_Contact.FirstName = txtFirstName.Text;
_Contact.LastName = txtLastName.Text;
_Contact.Email = txtEmail.Text;
_Contact.Phone = txtPhone.Text;
_Contact.Address = txtAddress.Text;
_Contact.DateOfBirth = dtpDateOfBirth.Value;
_Contact.CountryID = CountryID;

if (pictureBox1.ImageLocation!=null) {

_Contact.ImagePath= pictureBox1.ImageLocation;
} else {
_Contact.ImagePath = "";
}

if (_Contact.Save())
{
MessageBox.Show("Data Saved Successfully.");
}
else {
MessageBox.Show("Error: Data Didn't Save Successfully.");
}

_Mode = enMode.Update;

lblMode.Text = "Edit Contact ID = " + _Contact.ID;
lblContactID.Text = _Contact.ID.ToString();
}
```

وكده نبقي خلصنا وده الكود كامل بتاع الفورم

```
using ContactsBusinessLayer;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Contacts
{
public partial class frmAddEditContact : Form
{

enum enMode {AddNew=0,Update=1 };
enMode _Mode;

int _ContactID;
clsContact _Contact;

private void _FillCountriesInComboBox() {

DataTable dt= clsCountry.GetAllCountries();

foreach (DataRow row in dt.Rows) {

cbCountry.Items.Add(row["CountryName"]);
}

}

private void _LoadData() {

_FillCountriesInComboBox();
cbCountry.SelectedIndex = 0;

if (_Mode==enMode.AddNew) {
lblMode.Text = "Add New Contact";
_Contact=new clsContact();
return;
}

_Contact = clsContact.Find(_ContactID);

if (_Contact ==null) {

MessageBox.Show("This Form will be closed because no contact with ID = "+_ContactID);
this.Close();
return;
}

lblMode.Text = "Edit Contact ID = " + _ContactID;
lblContactID.Text=_ContactID.ToString();

txtFirstName.Text= _Contact.FirstName;
txtLastName.Text= _Contact.LastName;
txtEmail.Text= _Contact.Email;
txtPhone.Text= _Contact.Phone;
txtAddress.Text= _Contact.Address;
dtpDateOfBirth.Value = _Contact.DateOfBirth;

if (_Contact.ImagePath !="") {
```

```
pictureBox1.Load(_Contact.ImagePath);
}

llRemoveImage.Visible = (_Contact.ImagePath !="");
cbCountry.SelectedIndex = cbCountry.FindString( clsCountry.Find(_Contact.CountryID).CountryName);

}



public frmAddEditContact(int ID)
{
InitializeComponent();

this._ContactID = ID;

if (ID==-1) {
this._Mode = enMode.AddNew;
} else {
this._Mode = enMode.Update;
}
}

private void btnClose_Click(object sender, EventArgs e)
{
this.Close();
}

private void frmAddEditContact_Load(object sender, EventArgs e)
{
_LoadData();
}

private void llOpenFileDialog_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
openFileDialog1.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.gif;*.bmp";
openFileDialog1.FilterIndex = 1;
openFileDialog1.RestoreDirectory = true;

if (openFileDialog1.ShowDialog()==DialogResult.OK) {

string selectedFilePath=openFileDialog1.FileName;
pictureBox1.Load(selectedFilePath);
llRemoveImage.Visible = true;
}
}

private void llRemoveImage_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
pictureBox1.ImageLocation = null;
llRemoveImage.Visible = false;
}

private void btnSave_Click(object sender, EventArgs e)
{
int CountryID = clsCountry.Find(cbCountry.Text).ID;

_Contact.FirstName = txtFirstName.Text;
_Contact.LastName = txtLastName.Text;
_Contact.Email = txtEmail.Text;
_Contact.Phone = txtPhone.Text;
_Contact.Address = txtAddress.Text;
_Contact.DateOfBirth = dtpDateOfBirth.Value;
_Contact.CountryID = CountryID;

if (pictureBox1.ImageLocation!=null) {

_Contact.ImagePath= pictureBox1.ImageLocation;
} else {
_Contact.ImagePath = "";
```

```
}

if (_Contact.Save())
{
MessageBox.Show("Data Saved Successfully.");
}
else {
MessageBox.Show("Error: Data Didn't Save Successfully.");
}

_Mode = enMode.Update;

lblMode.Text = "Edit Contact ID = " + _Contact.ID;
lblContactID.Text = _Contact.ID.ToString();
}
}
}
```

What is Datatable?

هنا هنبدأ نتكلم عن ال datatable اللي استخدمناه في المشروع

ال data table اعتبره two dimentional array او جدول بتقدر تعرف الاعمده والصفوف اللي فيه وتعمل عليهم بحث وفلتر وترتيب ومش لازم تربطه بداتابيز

و هوا عباره عن data structure و data container

كنا حكينا في الكورس ده قبل كده انه فيه عندنا data reader وdata set ال data reader خلاص عرفناها و اتعاملنا معاها

ال data set كانت بتاخد الداتا من ال data reader عن طريق adapter وتخزنه وتخليك تعمل العمليات اللي انت عايز تعملها بس offline وبعد ماتخلص خالص بتروح تحط الداتا كلها علي بعضها تاني في الداتابيز

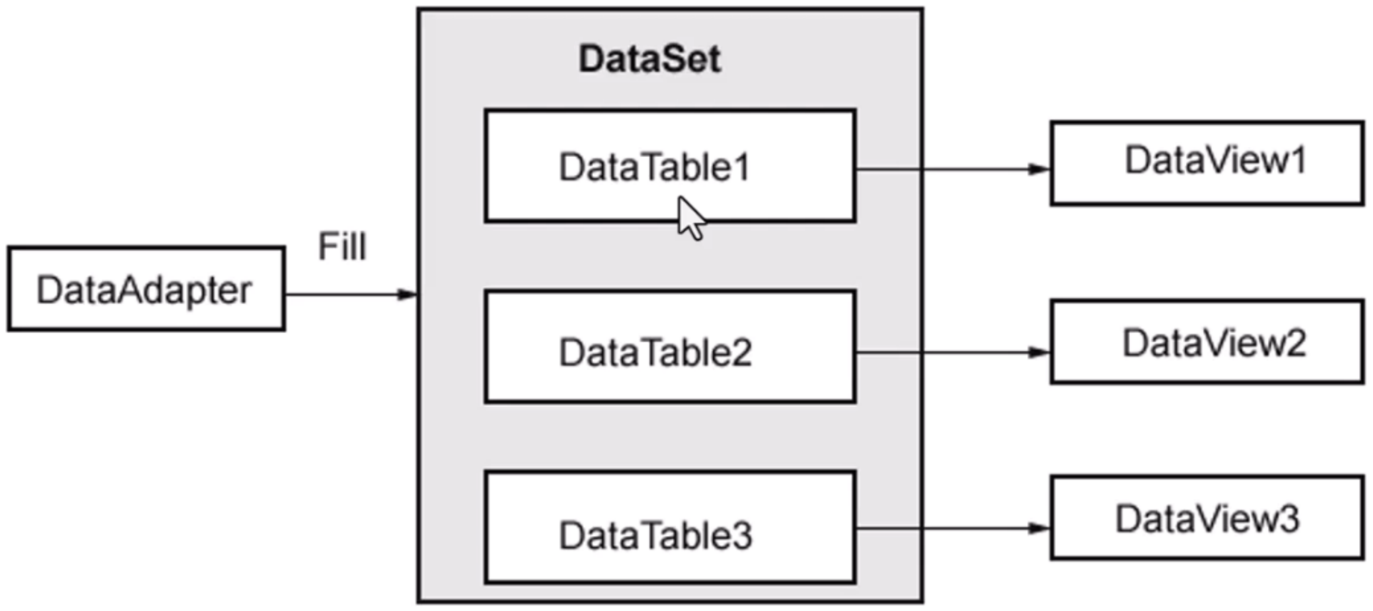
و قولنا انه ال data reader اسرع من ال dataset

هنا بيقولك انه ال data set هيا عباره عن مجموعه من ال data tables اللي بينهم علاقات

يعني اعتبرهم داتابيز بس اوفلاين

فيه حاجتين تانيه هنشرحهم بعدين و هما ال data adapter وال dataview

What is Datatable?



What is DataTable?

In C#, `DataTable` is a class provided by the .NET Framework that represents an in-memory table of data. It is part of the ADO.NET library and is used to store and manipulate tabular data. A `DataTable` can hold multiple rows and columns, similar to a database table or a spreadsheet.

Here are some key features and concepts related to `DataTable`:

1. Rows and Columns: A `DataTable` consists of a collection of rows and columns. Each row represents a record or a set of related data, and each column represents a specific data attribute or field.

2. Data Types: `DataTable` allows you to define the data types for each column, such as integers, strings, dates, etc. This ensures type safety and enables data validation.

3. Primary Key and Constraints: You can specify a primary key for a `DataTable` to enforce uniqueness and identify individual rows. Additionally, you can define constraints, such as unique constraints or foreign key constraints, to maintain data integrity.

4. Adding and Manipulating Data: You can add rows to a `DataTable` and populate them with data using the `NewRow` method. Columns can be accessed by their names or indexes, and data can be retrieved, modified, or deleted as needed.

5. Data Binding: `DataTable` supports data binding, allowing you to bind it to UI controls such as grids, lists, or combo boxes. This enables you to display and interact with the data in a user interface.

6. Querying and Sorting: You can perform various operations on a `DataTable` to query and manipulate the data it contains. These include filtering rows based on specific conditions, sorting the data based on column values, and performing aggregate calculations like sum, count, or average.

7. Serialization: `DataTable` can be serialized and deserialized to transfer or persist the data across different processes or systems. It can be stored in XML format or binary format using serialization techniques provided by .NET.

`DataTable` provides a rich set of methods and properties for working with tabular data. It is widely used in applications that involve data manipulation, data analysis, or data presentation scenarios.

DataTable Example 1 (Create Offline Data Table and ListData)

تعالي نعرف ازاي نعمل data table ونعرف فيه ال rows وال columns

تعالي نعمل console app

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
}
}
}
```

هنستخدم المكتبات دي

```
using System;
using System.Data;
using System.Linq;
```

مكتبة ال Data دي عشان موجود فيها ال data table

مكتبة ال Linq دي اللي هيا بتاعت ال max وال min والحوارات دي

طيب هناخد object من ال data table عادي

وعشان نضيف عمود جديد بستدعي method اسمها add موجوده في ال EmployeesDataTable.Columns ودي بتاخد اسم العمود وال data type بتاعه

طيب عاوزين نضيف records يعني صفوف

بتستدعي نفس ال method بس موجوده المرادي في ال EmployeesDataTable.Rows ودي بتاخد القيم اللي عاوز تحطها بترتيب الاعمده

```
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);
```

طيب عاوزين نمشي عال records دي ونطبع كل الداتا اللي في الجدول

قالك بتمشي عليهم ب for each وبتاخد object من data row وبعدين in وبعدين اسم الجدول وبعديه كلمة Rows

كده هيلف علي كل row ويخزنه في ال object اللي اسمه row وتقدر تستدعي القيم اللي في الصف ده بطريقتين

اول طريقه انك تكتب رقم العمود وتاني طريقه وهيا المفضله وهيا انك تكتب اسم العمود عشن الكود بتاعك يكون مقروء

```
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {

Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row["ID"], row["Name"], row["Country"], row["Salary"], row["Date"]);

Console.WriteLine();

Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);

Console.WriteLine("\n\n");
}

Console.ReadKey();
}
```

DataTable Example 2 (Count, Sum, Avg, Min, Max)

هنستعمل ال methods بتاعت مكتبة ال linq علي ال datatable

قالك فيه function سمها compute موجوده جوه ال datatable دي بتاخد حاجتين اسم العمليه والشرط

اسم العمليه بيكون في شكل string وبتكتب فيه اسم العمليه كأنك بتستدعي function وجواها بتكتب العمود اللي عاوز تعمل عليه العمليه دي

زي مثلا "SUM(Salary)"

ولو مفيش شرط بتكتب string.Empty

واللي خارج من ال function بتحوله ل double او int

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine(EmployeesCount);
Console.WriteLine(TotalSalaries);
Console.WriteLine(AverageSalaries);
Console.WriteLine(MinSalaries);
Console.WriteLine(MaxSalaries);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {

Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row["ID"], row["Name"], row["Country"], row["Salary"], row["Date"]);

Console.WriteLine();

Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);

Console.WriteLine("\n\n");
}

Console.ReadKey();
}
}
}
```

DataTable Example 3 (Filter Data)

عايز اعرض الموظفين اللي من الأردن بس

الفلتر هوا شرط ممكن تعمله علي الداتا نفسها او علي ال AGGREGATE FUNCTIONS

عشان تعمل فلتر عالداتا نفسها بتاخد OBJECT من ال dataRow بس بتعرفه ك array وبعدين بتخزن فيه الداتا متفلتره وبعدين تمشي عليهم بال foreach

طيب هتجيب الداتا متفلتره ازاي؟

فيه function في ال datatable اسمها select بتديها string مكون من اسم العمود ويساوي والقيمه اللي عايز تفلتر عليها

```
DataRow[] ResultRows = EmployeesDataTable.Select("Country= 'Jordan' ");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

طيب لو عايز افلتر ال aggregate function ؟

شايف ال string اللي حطيناه جوه ال string ؟

بتاخده تحطه مكان ال string.empty

```
ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "Country= 'Jordan' "));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "Country= 'Jordan' "));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "Country= 'Jordan' "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "Country= 'Jordan' "));
```

طيب عايز افلتر علي مصر او الأردن

بتكتب or عادي

```
Console.WriteLine();
Console.WriteLine();

ResultRows = EmployeesDataTable.Select("Country= 'Jordan' or Country= 'Egypt' ");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "Country= 'Jordan'  or Country= 'Egypt' "));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "Country= 'Jordan'  or Country= 'Egypt' "));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "Country= 'Jordan' or Country= 'Egypt' "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "Country= 'Jordan' or Country= 'Egypt' "));

Console.WriteLine();
Console.WriteLine($"Employees Count : {EmployeesCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");
```

عاوز افلتر عال id

```
ResultRows = EmployeesDataTable.Select("ID=1");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "ID=1"));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "ID=1"));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "ID= 1 "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "ID= 1 "));

Console.WriteLine();
Console.WriteLine($"Employees Count : {ResultCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");
```

وده الكود كله

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
```

```
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
Console.WriteLine($"Employees Count : {EmployeesCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");

Console.WriteLine();
Console.WriteLine();

//filter to jordan only

int ResultCount = 0;
DataRow[] ResultRows = EmployeesDataTable.Select("Country= 'Jordan' ");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "Country= 'Jordan' "));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "Country= 'Jordan' "));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "Country= 'Jordan' "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "Country= 'Jordan' "));

Console.WriteLine();
Console.WriteLine($"Employees Count : {ResultCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");
Console.WriteLine();
Console.WriteLine();


//filter to jordan or eygpt
ResultRows = EmployeesDataTable.Select("Country= 'Jordan' or Country= 'Egypt' ");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "Country= 'Jordan'  or Country= 'Egypt' "));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "Country= 'Jordan'  or Country= 'Egypt'
"));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "Country= 'Jordan' or Country= 'Egypt' "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "Country= 'Jordan' or Country= 'Egypt' "));

Console.WriteLine();
Console.WriteLine($"Employees Count : {ResultCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");

//filter to id=1
ResultRows = EmployeesDataTable.Select("ID=1");

foreach (DataRow row in ResultRows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

ResultCount = ResultRows.Count();
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)", "ID=1"));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)", "ID=1"));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)", "ID= 1 "));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)", "ID= 1 "));

Console.WriteLine();
Console.WriteLine($"Employees Count : {ResultCount}");
Console.WriteLine($"Total Salaries : {TotalSalaries}");
Console.WriteLine($"Average Salaries : {AverageSalaries}");
Console.WriteLine($"Min Salaries : {MinSalaries}");
Console.WriteLine($"Max Salaries : {MaxSalaries}");
Console.WriteLine();
Console.WriteLine();

Console.ReadKey();
}
}
}
```

Datatable Example 4 (Sorting)

ال sorting في ال data table بيكون بطئ هناخد بعدين حاجه اسرع

عشان تعمل sorting بتستدعي sort ودي موجوده في ال default view اللي موجود في ال datatable وبتاخد string عباره عن اسم العمود وتصاعدي او تنازلي

وبعد كده بتستدعي function اسمها to table موجوده في ال default view برضه واللي خارج منها بتخزنه مكان ال data tabe القديم

عايز ارتب حسب ال id تنازلي

```
EmployeesDataTable.DefaultView.Sort = "ID Desc";
EmployeesDataTable=EmployeesDataTable.DefaultView.ToTable();

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

عايز ارتب حسب الاسم تصاعدي

```
EmployeesDataTable.DefaultView.Sort = "Name ASC";
EmployeesDataTable = EmployeesDataTable.DefaultView.ToTable();

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

وده الكود كله

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
```

```
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
// sort id desc

EmployeesDataTable.DefaultView.Sort = "ID Desc";
EmployeesDataTable=EmployeesDataTable.DefaultView.ToTable();

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

// sort name asc
Console.WriteLine();
Console.WriteLine();

EmployeesDataTable.DefaultView.Sort = "Name ASC";
EmployeesDataTable = EmployeesDataTable.DefaultView.ToTable();

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.ReadKey();
}
}
}
```

Datatable Example 5 (Delete Rows)

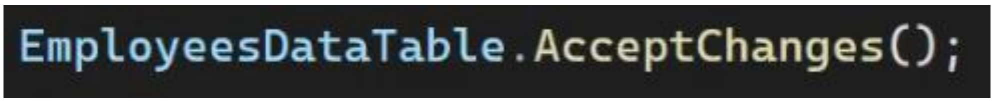
عاوزين نحذف صف من الجدول

يدوب بتفلتر الداتا عالصف او الصفوف اللي انت عاوزها وبتستدعي delete

عاوزين نحذف ال صف اللي الid بتاعه ب 4

```
DataRow[] Resultrows = EmployeesDataTable.Select("ID=4");
foreach (DataRow row in Resultrows)
{
row.Delete();
}

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

هنا بيقولك لو انت في DATASET وبتتعامل مع الداتا بيز بتكتب السطر ده عشان بعد الحذف يروح يعدل عالداتابيز



```
EmployeesDataTable.AcceptChanges();
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
// delete row id =4
DataRow[] Resultrows = EmployeesDataTable.Select("ID=4");
foreach (DataRow row in Resultrows)
{
row.Delete();
}
EmployeesDataTable.AcceptChanges();
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}


Console.ReadKey();
}
}
}
```

Datatable Example 6 (Update Rows)

عشان تعمل update ل records معينه بتعمل عليها فلتر وبعدين بتعدلها وتعمل

accept changes لو متوصل بداتابيز

عاوزين نعدل عال record اللي ال id بتاعه ب 4

```
DataRow[] Resultrows = EmployeesDataTable.Select("ID=4");
foreach (DataRow row in Resultrows)
{
row["Name"]="Maha Ahmed";
row["Salary"]=900;
}
EmployeesDataTable.AcceptChanges();
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
// update row id =4
DataRow[] Resultrows = EmployeesDataTable.Select("ID=4");
foreach (DataRow row in Resultrows)
{
row["Name"]="Maha Ahmed";
row["Salary"]=900;
}
EmployeesDataTable.AcceptChanges();
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}


Console.ReadKey();
}
}
}
```

Datatable Example 7 (Clear)

Clear All Data:

To clear all data (delete all records in the Datatable) you simply use .Clear method.

//Clear all Data

EmployeesDataTable.Clear();

لو عايز تحذف الداتا كلها بتستدعي الmethod اللي اسمها clear

```
EmployeesDataTable.Clear();
EmployeesDataTable.AcceptChanges();
```

```
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
EmployeesDataTable.Clear();
EmployeesDataTable.AcceptChanges();
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}


Console.ReadKey();
}
}
}
```

DataTable Example 8 (Primary Key)

ال primary key ساعات بيكون عمود واحد او عدة اعمده

زي مافيه Data Row فيه برضه DataColumn

طيب عشان اعمل primary key

بعرف array من ال data column وال constructor بتاعه بياخد عدد الاعمده

بعدين بتخزن فيه العمود اللي انت عايزه

بعدين بتستدعي primary key موجود في ال datatable وبتديله ال data column اللي انت عملته

```
DataColumn[] PrimaryKey = new DataColumn[1];
PrimaryKey[0] = EmployeesDataTable.Columns["ID"];
EmployeesDataTable.PrimaryKey = PrimaryKey;
```

جرب تكرر ال ID هيضرب منك

ماتقدرش تعمل RELATION بين الجدوال الا لما تجمعهم وتحطهم في data set

ماينفعش تضيف ال PRIMARY KEY بالطريقه دي

```
EmployeesDataTable.PrimaryKey =EmployeesDataTable.Columns[0];
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name",typeof(string));
EmployeesDataTable.Columns.Add("Country",typeof(string));
EmployeesDataTable.Columns.Add("Salary",typeof (Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1,"Mohammed Abu-Hadhoud","Jordan",5000,DateTime.Now);
EmployeesDataTable.Rows.Add(2,"Ali Maher","KSA",525.5,DateTime.Now);
EmployeesDataTable.Rows.Add(3,"Lina Kamal","Jordan",730.5,DateTime.Now);
EmployeesDataTable.Rows.Add(4,"Fadi Jameel","Egypt",800,DateTime.Now);
EmployeesDataTable.Rows.Add(5,"Omar Mahmoud","Lebanon",7000,DateTime.Now);

int EmployeesCount = 0;
double TotalSalaries = 0;
double AverageSalaries = 0;
double MinSalaries = 0;
double MaxSalaries = 0;

//get all employees
EmployeesCount= EmployeesDataTable.Rows.Count;
TotalSalaries = Convert.ToDouble(EmployeesDataTable.Compute("SUM(Salary)",String.Empty));
AverageSalaries = Convert.ToDouble(EmployeesDataTable.Compute("AVG(Salary)",String.Empty));
MinSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Min(Salary)",String.Empty));
MaxSalaries = Convert.ToDouble(EmployeesDataTable.Compute("Max(Salary)",String.Empty));

DataColumn[] PrimaryKey = new DataColumn[1];
PrimaryKey[0] = EmployeesDataTable.Columns["ID"];
EmployeesDataTable.PrimaryKey = PrimaryKey;



Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine();
EmployeesDataTable.Clear();
EmployeesDataTable.AcceptChanges();
foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}


Console.ReadKey();
}
}
}
```

DataTable Example 9 (Autoincrement and Others)

عشان تقدر تتحكم بالخصائص بتاعت العمود محتاج تعرفه بطريقه تانيه ويا انك تاخد object من data column وبعدين تديله الخصائص

ال name ده الاسم البرمجي

ال caption ده الاسم اللي هيظهر بيه

```
DataColumn dtColumn= new DataColumn();
dtColumn.DataType = typeof(int);
dtColumn.ColumnName = "ID";
dtColumn.AutoIncrement = true;
dtColumn.AutoIncrementSeed = 1;
dtColumn.AutoIncrementStep = 1;
dtColumn.Caption = "Employee ID";
dtColumn.ReadOnly = true;
dtColumn.Unique = true;

EmployeesDataTable.Columns.Add(dtColumn);
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
```

```
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();

DataColumn dtColumn= new DataColumn();
dtColumn.DataType = typeof(int);
dtColumn.ColumnName = "ID";
dtColumn.AutoIncrement = true;
dtColumn.AutoIncrementSeed = 1;
dtColumn.AutoIncrementStep = 1;
dtColumn.Caption = "Employee ID";
dtColumn.ReadOnly = true;
dtColumn.Unique = true;

EmployeesDataTable.Columns.Add(dtColumn);
//////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(string);
dtColumn.ColumnName = "Name";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Name";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
//////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(string);
dtColumn.ColumnName = "Country";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Country";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
//////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(double);
dtColumn.ColumnName = "Salary";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Salary";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
//////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(DateTime);
dtColumn.ColumnName = "Date";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Date";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);

DataColumn[] PrimaryKey = new DataColumn[1];
PrimaryKey[0] = EmployeesDataTable.Columns["ID"];
EmployeesDataTable.PrimaryKey = PrimaryKey;
//////////////////////////////////////////////////
EmployeesDataTable.Rows.Add(null, "Mohammed Abu-Hadhoud", "Jordan", 5000, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Ali Maher", "KSA", 525.5, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Lina Kamal", "Jordan", 730.5, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Fadi Jameel", "Egypt", 800, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Omar Mahmoud", "Lebanon", 7000, DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}




Console.ReadKey();
}
}
}
```

What is Dataview?

ال data view دهبيكون زي طريقه عرض للداتا اللي في ال data table وهوا بيكون زي

snap shot او لقطه من الداتا

انت اعتبره سكرتير لل data table

وكل data table بيكون ليه data view بيكون ال default بتاعه

لكن طبعا ممكن تعمل اكتر من data view ل data table واحد بحيث انك تيجي في كل view تعرض مجموعه معينه من الداتا

الداتا اللي بتتعدل عن طريق ال data view بتتعدل تلقائيا في ال datatable والعكس وهوا بيقدم الخصائص بتاعت ال filter وال sort والتعديل

طيب ماانا مستريح مع ال data table ليه اشتغل مع السكرتير بتاعه؟

هنا بيقولك انه ال algorihms اللي موجوده في ال data view بتاعت الترتيب والبحث والكلام ده هيا اسرع بكتر من اللي موجوده في ال data table عشان كده يفضل انك تتعامل معاها

وهيا مش بتخزن داتا هيا بتعرض الداتا وبتكون read only واخف من ال data table

In C#, both `DataView` and `DataTable` are classes provided by the .NET Framework for working with tabular data. While they serve similar purposes, there are some differences between them, and the choice between `DataView` and `DataTable` depends on your specific requirements. Here are a few reasons why you might choose to use `DataView` over `DataTable`:

1. Data Manipulation: `DataView` provides more flexible and efficient data manipulation capabilities compared to `DataTable`. It allows you to apply filters, sort and search data, and perform custom data projections using LINQ queries. These operations are often more convenient and performant with `DataView`.

2. Lightweight: `DataView` is a lightweight wrapper around a `DataTable` that provides a read-only, customized view of the data. It does not duplicate the underlying data but instead provides a flexible way to access and manipulate it. If you only need to query or display data without making any modifications, using `DataView` can be more memory-efficient.

3. Sorting and Filtering: `DataView` makes it easy to sort and filter data using its built-in functionalities. You can specify multiple sort criteria, apply complex filters using

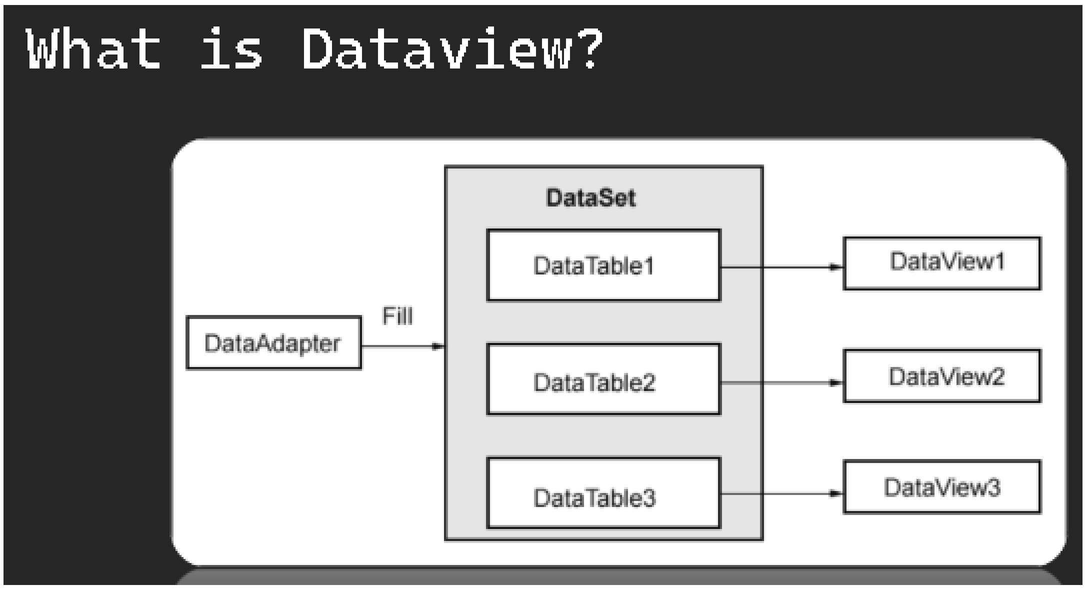
expressions, and even create custom views based on specific criteria. `DataTable` also supports sorting and filtering, but `DataView` provides a more convenient and expressive syntax for these operations.

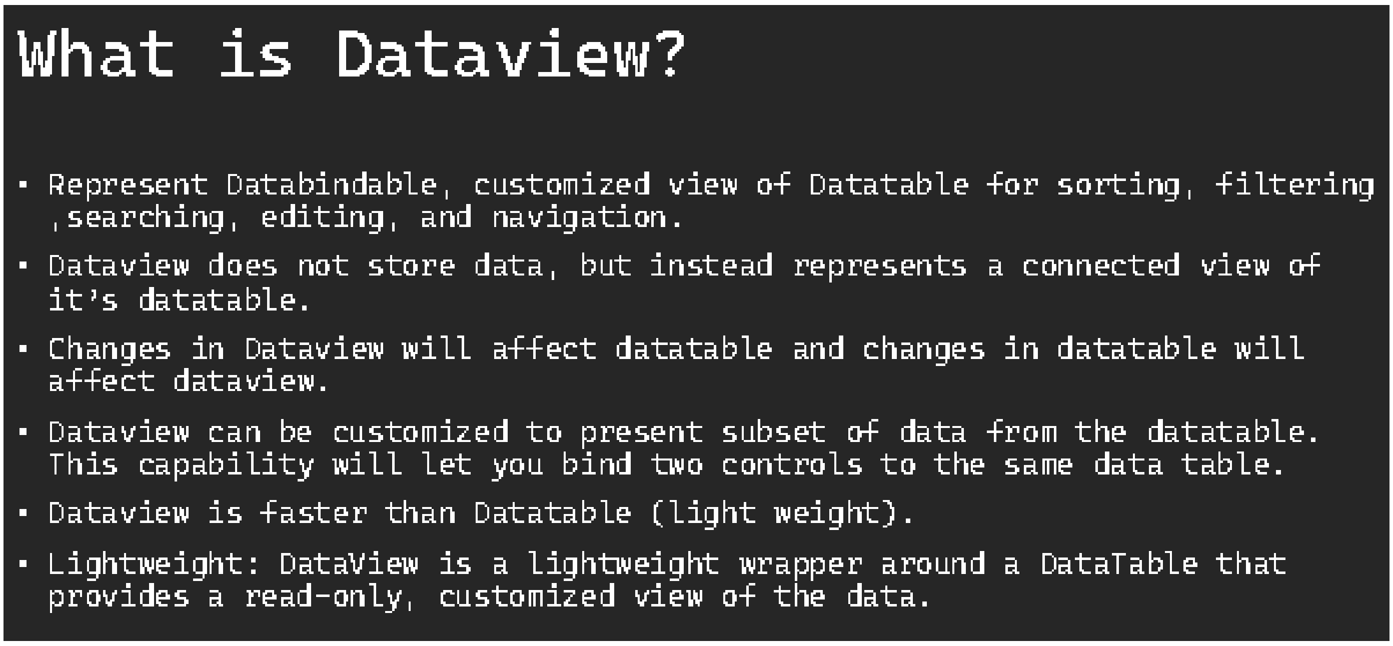
4. Data Binding: If you are working with data binding scenarios, `DataView` provides more control and flexibility. It allows you to bind the view to UI controls and automatically update the display when the underlying data changes. `DataTable` can also be bound to controls, but `DataView` offers more advanced features in this regard.

5. Performance: In some cases, `DataView` can offer better performance compared to `DataTable`. Since `DataView` provides a customized view of the data, it can optimize operations like filtering and sorting, resulting in faster execution times. If you frequently perform these operations on large datasets, `DataView` might provide better performance.

However, it's worth noting that `DataTable` has its advantages as well. It is a more robust and comprehensive data structure that can hold multiple tables, enforce constraints, and support more complex data operations. If you require these advanced features or need to perform extensive modifications to the data structure itself, `DataTable` might be a better choice.

In summary, `DataView` is generally preferred when you need a lightweight, customizable view of data for querying, sorting, and filtering, while `DataTable` is suitable for scenarios that involve complex data operations, constraints, and multiple tables.





What is Dataview?

- Represent Databindable, customized view of Datatable for sorting, filtering ,searching, editing, and navigation.
- Dataview does not store data, but instead represents a connected view of it's datatable.
- Changes in Dataview will affect datatable and changes in datatable will affect dataview.
- Dataview can be customized to present subset of data from the datatable. This capability will let you bind two controls to the same data table.
- Dataview is faster than Datatable (light weight).
- Lightweight: DataView is a lightweight wrapper around a DataTable that provides a read-only, customized view of the data.

Create Dataview from Datatable

عشان تعمل data view بتعمل object من data view وبعدين تستدعي ال default view اللي في ال data table وتخزنه فيه

عشان تلف علي كل العناصر بتلف عليهم ب for loop وبتتعامل معاه كأنه two dimentional array

```
DataView EmployeesDataView1 = EmployeesDataTable.DefaultView;
Console.WriteLine();
for (int i=0; i<EmployeesDataView1.Count;i++) {
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}
```

Filtering Data in Dataview

عشان تعمل فلتر عالداتا اللي جايه من ال data view بتستدعي Row Filter موجود في ال data view

```
EmployeesDataView1.RowFilter = "Country='Jordan' or Country ='Egypt' ";
for (int i = 0; i < EmployeesDataView1.Count; i++)
{
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}
```

Sorting Data in Dataview

عشان ترتب الداتا بتستدعي sort من الdata view

```
//////////////////////sort data view
Console.WriteLine();
```

```
EmployeesDataView1.Sort = "Name ASC ";
for (int i = 0; i < EmployeesDataView1.Count; i++)
{
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}
```

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();

DataColumn dtColumn= new DataColumn();
dtColumn.DataType = typeof(int);
dtColumn.ColumnName = "ID";
dtColumn.AutoIncrement = true;
dtColumn.AutoIncrementSeed = 1;
dtColumn.AutoIncrementStep = 1;
dtColumn.Caption = "Employee ID";
dtColumn.ReadOnly = true;
dtColumn.Unique = true;

EmployeesDataTable.Columns.Add(dtColumn);
/////////////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(string);
dtColumn.ColumnName = "Name";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Name";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
/////////////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(string);
dtColumn.ColumnName = "Country";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Country";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
/////////////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(double);
dtColumn.ColumnName = "Salary";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Salary";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);
/////////////////////////////////////////////////
dtColumn = new DataColumn();
dtColumn.DataType = typeof(DateTime);
dtColumn.ColumnName = "Date";
dtColumn.AutoIncrement = false;
dtColumn.Caption = "Date";
dtColumn.ReadOnly = false;
dtColumn.Unique = false;

EmployeesDataTable.Columns.Add(dtColumn);

DataColumn[] PrimaryKey = new DataColumn[1];
PrimaryKey[0] = EmployeesDataTable.Columns["ID"];
EmployeesDataTable.PrimaryKey = PrimaryKey;
////////////////////////////////////////////////////
EmployeesDataTable.Rows.Add(null, "Mohammed Abu-Hadhoud", "Jordan", 5000, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Ali Maher", "KSA", 525.5, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Lina Kamal", "Jordan", 730.5, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Fadi Jameel", "Egypt", 800, DateTime.Now);
EmployeesDataTable.Rows.Add(null, "Omar Mahmoud", "Lebanon", 7000, DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows) {
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

//////////////////////////////////
DataView EmployeesDataView1 = EmployeesDataTable.DefaultView;
Console.WriteLine();
for (int i=0; i<EmployeesDataView1.Count;i++) {
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}
///////////////////////filter data view
Console.WriteLine();
EmployeesDataView1.RowFilter = "Country='Jordan' or Country ='Egypt' ";
for (int i = 0; i < EmployeesDataView1.Count; i++)
{
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}
//////////////////////sort data view
Console.WriteLine();

EmployeesDataView1.Sort = "Name ASC ";
for (int i = 0; i < EmployeesDataView1.Count; i++)
{
Console.WriteLine("{0},{1},{2},{3},{4}",
EmployeesDataView1[i][0], EmployeesDataView1[i][1],
EmployeesDataView1[i][2], EmployeesDataView1[i][3], EmployeesDataView1[i][4]);
}

Console.ReadKey();
}
}
}
```

What is Dataset?

ال DATA SET هوا container بتقدر تخزن فيه مجموعه من ال data table وبيعتبر نسخه اوفلاين من الداتابيز وبعد كده تقعد تعمل العمليات اللي انت عايزها عليها بدل ال query والكلام ده بس هتكون بطيئه الأفضل ماتستخدمهاش

What is Dataset?

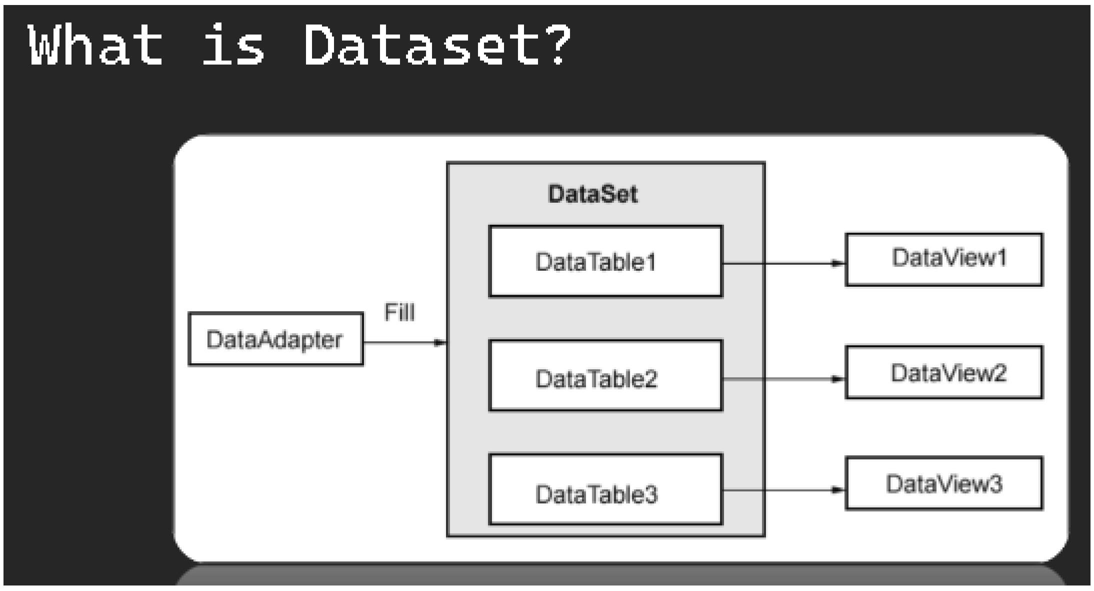
In C#, a dataset is an in-memory representation of a collection of data that can be used for storing and manipulating structured data. It is a part of the ADO.NET technology stack, which is used for data access in .NET applications.

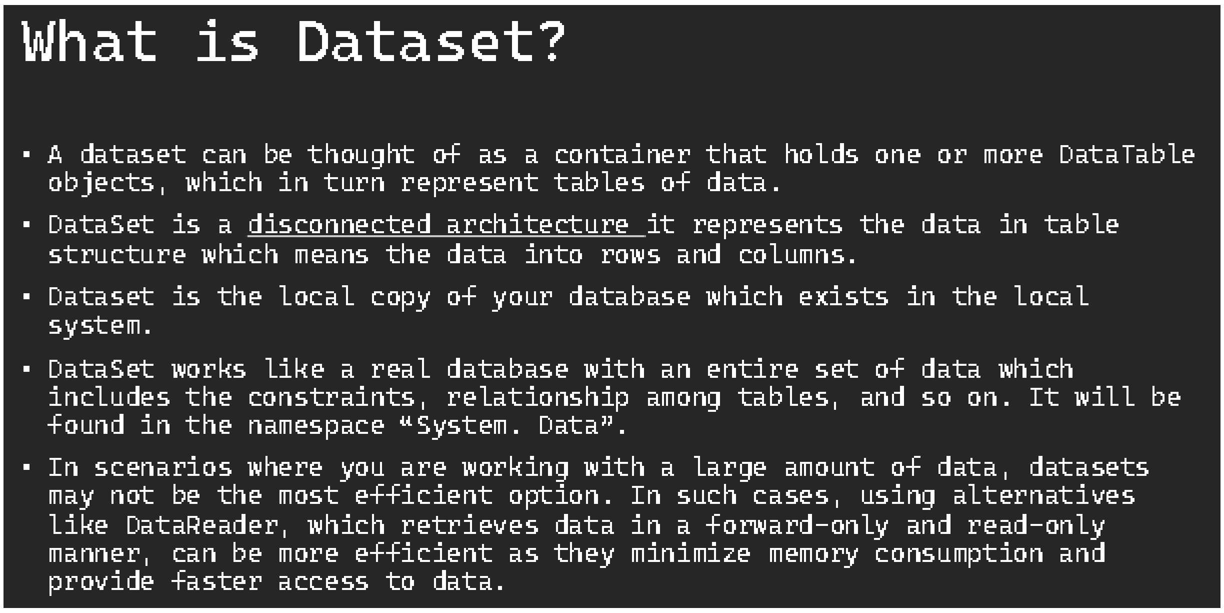
A dataset can be thought of as a container that holds one or more DataTable objects, which in turn represent tables of data. Each DataTable within a dataset contains DataColumn objects that define the structure and data types of the columns, as well as DataRow objects that represent the actual data rows.

Datasets provide a disconnected, in-memory representation of data, meaning that they can be filled with data from a data source (such as a database) and then disconnected from the data source. This allows for offline manipulation and analysis of the data without constantly being connected to the original data source.

Compared to other data access methods in C#, such as using a DataReader , datasets can sometimes be slower due to their inherent overhead. Datasets store data in memory in a disconnected manner, which means that data needs to be loaded from a data source into memory and then synchronized back to the data source when changes are made. This synchronization process can introduce additional overhead and impact performance.

In scenarios where you are working with a large amount of data, datasets may not be the most efficient option. In such cases, using alternatives like DataReader, which retrieves data in a forward-only and read-only manner, can be more efficient as they minimize memory consumption and provide faster access to data.





What is Dataset?

- A dataset can be thought of as a container that holds one or more DataTable objects, which in turn represent tables of data.
- DataSet is a disconnected architecture it represents the data in table structure which means the data into rows and columns.
- Dataset is the local copy of your database which exists in the local system.
- DataSet works like a real database with an entire set of data which includes the constraints, relationship among tables, and so on. It will be found in the namespace "System. Data".
- In scenarios where you are working with a large amount of data, datasets may not be the most efficient option. In such cases, using alternatives like DataReader, which retrieves data in a forward-only and read-only manner, can be more efficient as they minimize memory consumption and provide faster access to data.

Create Dataset

هنعمل جدولين ونضيفهم لل dataset ونطبعهم

دول الجدولين (في ال constructor تقدر تكتب اسم للجدول ولما تيجي تستدعيه هتستدعيه عن طريق رقم)

```
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name", typeof(string));
EmployeesDataTable.Columns.Add("Country", typeof(string));
EmployeesDataTable.Columns.Add("Salary", typeof(Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1, "Mohammed Abu-Hadhoud", "Jordan", 5000, DateTime.Now);
EmployeesDataTable.Rows.Add(2, "Ali Maher", "KSA", 525.5, DateTime.Now);
EmployeesDataTable.Rows.Add(3, "Lina Kamal", "Jordan", 730.5, DateTime.Now);
EmployeesDataTable.Rows.Add(4, "Fadi Jameel", "Egypt", 800, DateTime.Now);
EmployeesDataTable.Rows.Add(5, "Omar Mahmoud", "Lebanon", 7000, DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

DataTable DeaertmentDataTable=new DataTable();
DeaertmentDataTable.Columns.Add("ID", typeof(int));
DeaertmentDataTable.Columns.Add("Name", typeof(string));

DeaertmentDataTable.Rows.Add(1, "Mrketing");
DeaertmentDataTable.Rows.Add(2, "IT");
DeaertmentDataTable.Rows.Add(3, "HR");

Console.WriteLine("\nDepartment List\n");
foreach (DataRow row in DeaertmentDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
```

بعدين بتضيف ال data table اللي عملتهم لل data set عن طريق tables.add

اعتبر ال dataset عباره عن array of tables

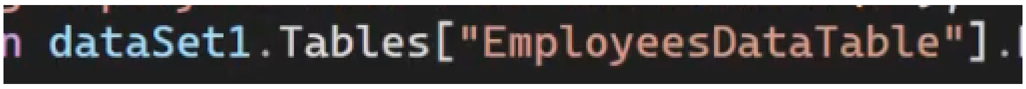
```
DataSet dataSet1= new DataSet();
dataSet1.Tables.Add(EmployeesDataTable);
dataSet1.Tables.Add(DepaertmentDataTable);

Console.WriteLine("\nEmployees List FROM DATA SET\n");

foreach (DataRow row in dataSet1.Tables[0].Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine("\nDepartment List FROM DATA SET\n");
foreach (DataRow row in dataSet1.Tables[1].Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
```

وده الكود كله

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable();
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name", typeof(string));
EmployeesDataTable.Columns.Add("Country", typeof(string));
EmployeesDataTable.Columns.Add("Salary", typeof(Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1, "Mohammed Abu-Hadhoud", "Jordan", 5000, DateTime.Now);
EmployeesDataTable.Rows.Add(2, "Ali Maher", "KSA", 525.5, DateTime.Now);
EmployeesDataTable.Rows.Add(3, "Lina Kamal", "Jordan", 730.5, DateTime.Now);
EmployeesDataTable.Rows.Add(4, "Fadi Jameel", "Egypt", 800, DateTime.Now);
EmployeesDataTable.Rows.Add(5, "Omar Mahmoud", "Lebanon", 7000, DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

DataTable DepaertmentDataTable=new DataTable();
DepaertmentDataTable.Columns.Add("ID", typeof(int));
DepaertmentDataTable.Columns.Add("Name", typeof(string));

DepaertmentDataTable.Rows.Add(1, "Mrketing");
DepaertmentDataTable.Rows.Add(2, "IT");
DepaertmentDataTable.Rows.Add(3, "HR");

Console.WriteLine("\nDepartment List\n");
foreach (DataRow row in DepaertmentDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
//////////////////////////////////////////////
DataSet dataSet1= new DataSet();
dataSet1.Tables.Add(EmployeesDataTable);
dataSet1.Tables.Add(DepaertmentDataTable);

Console.WriteLine("\nEmployees List FROM DATA SET\n");

foreach (DataRow row in dataSet1.Tables[0].Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine("\nDepartment List FROM DATA SET\n");
foreach (DataRow row in dataSet1.Tables[1].Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
Console.ReadKey();
}
}
}
```

Access Datatables Inside Dataset By Name

عشان تقدر تعمل الحركه دي



```
n dataSet1.Tables["EmployeesDataTable"].
```

بدل ما تكتب رقم الجدول

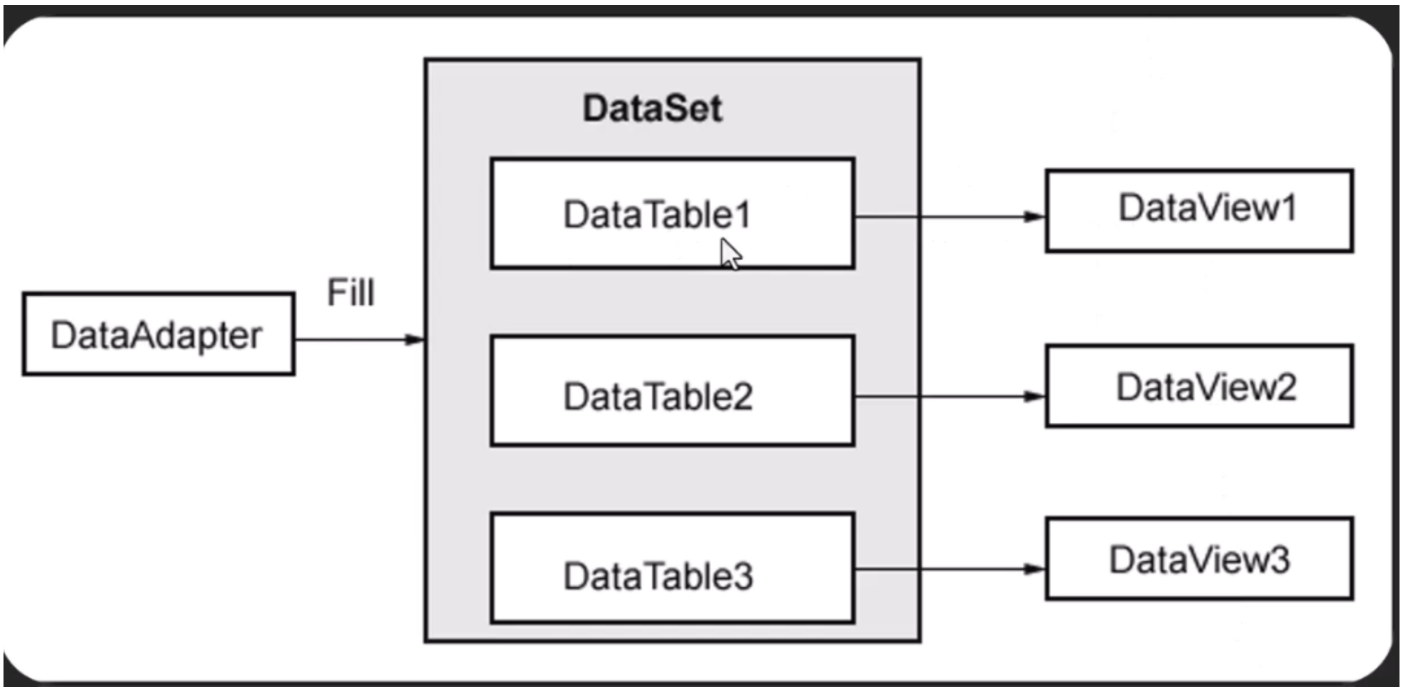
بتيجي في ال constructor بتاع ال data table وبتكتب فيه اسم الجدول

```
using System;
using System.Data;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
DataTable EmployeesDataTable = new DataTable("EmployeesDataTable");
EmployeesDataTable.Columns.Add("ID", typeof(int));
EmployeesDataTable.Columns.Add("Name", typeof(string));
EmployeesDataTable.Columns.Add("Country", typeof(string));
EmployeesDataTable.Columns.Add("Salary", typeof(Double));
EmployeesDataTable.Columns.Add("Date", typeof(DateTime));

EmployeesDataTable.Rows.Add(1, "Mohammed Abu-Hadhoud", "Jordan", 5000, DateTime.Now);
EmployeesDataTable.Rows.Add(2, "Ali Maher", "KSA", 525.5, DateTime.Now);
EmployeesDataTable.Rows.Add(3, "Lina Kamal", "Jordan", 730.5, DateTime.Now);
EmployeesDataTable.Rows.Add(4, "Fadi Jameel", "Egypt", 800, DateTime.Now);
EmployeesDataTable.Rows.Add(5, "Omar Mahmoud", "Lebanon", 7000, DateTime.Now);

Console.WriteLine("\nEmployees List\n");

foreach (DataRow row in EmployeesDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

DataTable DepaertmentDataTable=new DataTable("DepaertmentDataTable");
DepaertmentDataTable.Columns.Add("ID", typeof(int));
DepaertmentDataTable.Columns.Add("Name", typeof(string));

DepaertmentDataTable.Rows.Add(1, "Mrketing");
DepaertmentDataTable.Rows.Add(2, "IT");
DepaertmentDataTable.Rows.Add(3, "HR");

Console.WriteLine("\nDepartment List\n");
foreach (DataRow row in DepaertmentDataTable.Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
/////////////////////////////////////////////////
DataSet dataSet1= new DataSet();
dataSet1.Tables.Add(EmployeesDataTable);
dataSet1.Tables.Add(DepaertmentDataTable);

Console.WriteLine("\nEmployees List FROM DATA SET\n");

foreach (DataRow row in dataSet1.Tables["EmployeesDataTable"].Rows)
{
Console.WriteLine("ID: {0}\t Name: {1}\t Country: {2}\t Salary: {3}\t Date: {4}",
row[0], row[1], row[2], row[3], row[4]);
}

Console.WriteLine("\nDepartment List FROM DATA SET\n");
foreach (DataRow row in dataSet1.Tables["DepaertmentDataTable"].Rows)
{
Console.WriteLine("ID: {0}\t Department: {1}",
row[0], row[1]);
}
Console.ReadKey();
}
}
}
```

What is DataAdapter?

الموبايل بتاعك لو جيت وصلته بالكهرباء مباشرة هيولع في وشك فعشان كده عملوله محول او شاحن خاص الشاحن ده بيكون زي الكوبري بياخد الكهرباء بتاعت البيت وبيحولها لكهرباء يقدر الموبايل يتعامل معاها

هنا نفس الفكره بالنسبه لل data adapter هوا كوبري بيجيب الداتا من الداتابيز ويحطها في ال data set وده شغل الـmouse developer بس بناخده عشان نعرف الدنيا ماشيه فيه الزاي



What is DataAdapter?

• In C#, a DataAdapter is a class that acts as a bridge between a DataSet and a data source, such as a database. It provides methods for populating a DataSet with data from the data source and updating the data source with changes made to the DataSet.

What is DataAdapter?

In C#, a DataAdapter is a class that acts as a bridge between a DataSet and a data source, such as a database. It provides methods for populating a DataSet with data from the data source and updating the data source with changes made to the DataSet.

DataAdapter Example

هنستعمل الـhr database اللي في الكورس رقم 17 هناخد منها الداتا ونعرضها

هنعمل connection string و data set عشان نستقبل فيها الداتا و query

```
string ConnectionString = "Server=.;Database=HR_DB;User Id=sa;Password=sa123456";

DataSet dataSet = new DataSet();

string Query = "Select * from Employees";
```

بعد كده هنعرف الادابتر وال constructor بتاعه بياخد query و connection string

```
SqlDataAdapter dataAdapter = new SqlDataAdapter(Query, ConnectionString);
```

بعدين هنعرف connection ونفتحه

```
SqlConnection Connection=new SqlConnection(ConnectionString);
Connection.Open();
```

ال data adapter بيحتاج يتخزن فيه connection فهنديله ال connection اللي عندنا بس ال connection ده هنجيبه من ال select command عشان ال query اللي عندنا هيا امر select

```
dataAdapter.SelectCommand.Connection = Connection;
```

عشان نعبي الداتا في ال data set بنستدعي function اسمها fill موجوده في كلاس ال adapter

وبتاخد اسم ال dataset واسم للجدول

```
dataAdapter.Fill(dataSet,"Employees");
```

وبعدين بنقفل ال connection وبنعرف datatable نحط فيه الجدول اللي موجود في ال data set ونعرض كل الداتا اللي فيه

```
Connection.Close();
DataTable dt = dataSet.Tables["Employees"];
foreach (DataRow row in dt.Rows) {
Console.WriteLine("Customer ID: {0}, Name: {1}, LastName: {2}", row["ID"], row["FirstName"], row["LastName"]);
}
```

بتقعد بقي تعمل العمليات بتاعتك عن طريق ال view او عن طريق ال data table وبتعدين بتعمل update عشان ياخد الداتا اللي عندك بعد التعديل ويوديها للداتابيز تاني

فبتفتح الاتصال مع الداتابيز

```
Connection.Open ();
```

بعدين بستدعي ال connection اللي موجود في الادابتر بس المرادي عن طريق

update command وبحط فيها ال connection اللي عندي

```
Connection.Open ();

dataAdapter.UpdateCommand.Connection = Connection;
```

وبعدين بستدعي function اسمها update موجوده في ال data adapter ودي بتاخد dataset و اسم الجدول اللي هتعمله update

وبعدين تقفل الاتصال

```
dataAdapter.Update(dataSet,"Employees");


Connection.Close ();
```

بالنسبالي حصل exception في السطر ده

```
dataAdapter.UpdateCommand.Connection = Connection;
```

فلما دورت لقيت انه الحل عن طريق اني استدعي update command واخزن فيه command عادي بياخد query و connection واشتغل عادي

```
dataAdapter.UpdateCommand= new SqlCommand("UPDATE Employees SET FirstName = @FirstName, LastName = @LastName WHERE ID = @ID", Connection);
```

وده الكود كله

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Linq;


namespace DataTableExample1
{
internal class Program
{
static void Main(string[] args)
{
string ConnectionString = "Server=.;Database=HR_DB;User Id=sa;Password=sa123456";

DataSet dataSet = new DataSet();

string Query = "Select * from Employees";
SqlDataAdapter dataAdapter = new SqlDataAdapter(Query, ConnectionString);

SqlConnection Connection=new SqlConnection(ConnectionString);
Connection.Open();

dataAdapter.SelectCommand.Connection = Connection;
dataAdapter.Fill(dataSet,"Employees");

Connection.Close();
DataTable dt = dataSet.Tables["Employees"];
foreach (DataRow row in dt.Rows) {
Console.WriteLine("Customer ID: {0}, Name: {1}, LastName: {2}", row["ID"], row["FirstName"], row["LastName"]);
}

Connection.Open ();

// dataAdapter.UpdateCommand.Connection = Connection;
dataAdapter.UpdateCommand= new SqlCommand("UPDATE Employees SET FirstName = @FirstName, LastName =
@LastName WHERE ID = @ID", Connection);
dataAdapter.Update(dataSet,"Employees");


Connection.Close ();
Console.ReadKey();
}
}
}
```

How to be a Mouse Developer?

ماتشتغلش بال dataset

End Of Course

نصائح اسمعها من الفيديو احسن

واقعد طبق واعمل مشاريع

The End