



---

# الدواتا أكسيس لاير

# فهي السسي تتارپ

---

من إعداد : خالد السعداني



## بسم الله الرحمن الرحيم

وصلتني رسالة من عند أحد الإخوة الراغبين في التعرف أكثر على ماهية الداتا أكسيس لاير ، وعلى كيفية اشتغالها.فبدأت أعد له هذا الكتيب خصيصا له حتى يستمد منه ما هو محتاج إليه.

لكن بمجرد ما تقدمت في الكتيب، تساءلت لماذا لا أجعله في متناول الجميع حتى تعم الفائدة و يتسنى لأشخاص آخرين في نفس الموقف الظفر ولو بجزء قليل من المعرفة.

على كل حال فهذا الكتيب يأتي مكملا شيئا ما لكتاب "سبيلك المختصر لتعلم السي شارب" ، لأنني في الحقيقة كنت أنوي أن أوّلف جزء ثانيا يختص في قواعد البيانات و عمليات ربطها مع الفيچوال استوديو، ولكن نظرا لظروف شخصية لم أستطع البداية فعليا.وأتمنى من الله العلي القدير أن تتاح لي الفرصة لأنطلق مع الجزء الثاني.

ولا ننسى كما جرت العادة أن ننبهكم إلى أن "لكل شيء إذا ما تم نقصان" ، فأدنى خطأ تتمكنون من العثور عليه بين طيات الكتاب لا تتوانوا في إرشادنا إليه ولكم منا جزيل الشكر .

أتوقف عند هذا الحد حتى لا أكون ضيفا ثقيلا، و أتمنى لك -أيها القارئ- مسيرة موفقة في حياتك،  
واللهم انصر الإسلام و المسلمين !

خالد السعداني



## قبل أن تبدأ..

حتى تكون هناك استفادة كبيرة من هذا الكتيب، عليك أخي القارئ أن تكون مطلعاً  
على :

○ البرمجة بلغة السي شارب

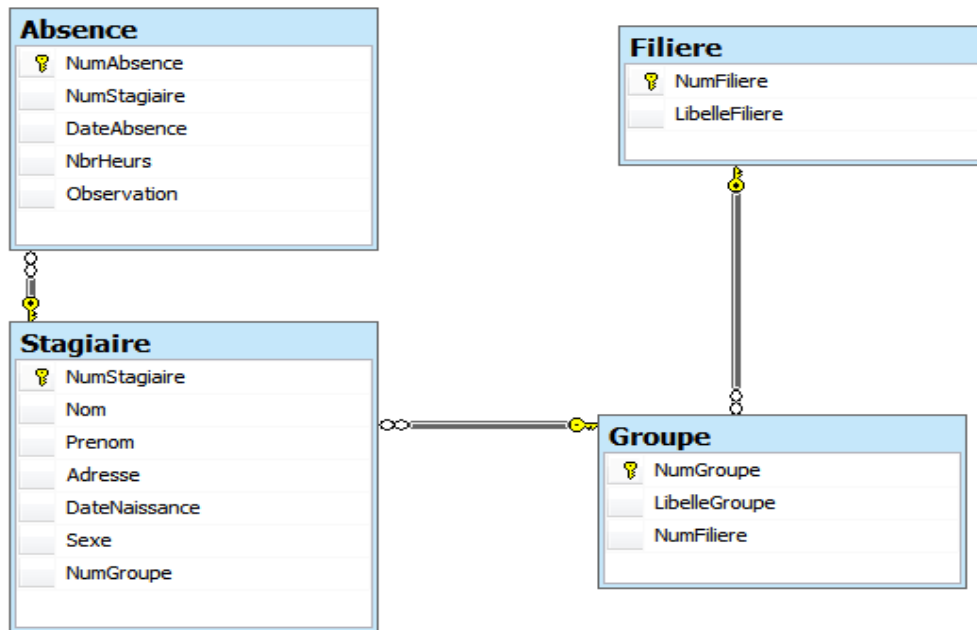
○ قواعد البيانات SQL Server



” يا أيها الذين آمنوا اتقوا الله و قولوا  
قولا سديدا. يصلح لكم أعمالكم و يغفر لكم  
ذنوبكم ومن يطع الله و رسوله فقد فاز  
□ فوزا عظيما ”

الأحزاب : 70 و 71

بالنسبة لقاعدة البيانات فهذه صورتها :



كما تلاحظ هناك أربعة جداول:

- **Stagiaire** : وهو جدول الطلاب و يضم الحقول التالية (رقم الطالب، الاسم، النسب، العنوان، تاريخ الازدياد، الجنس) أما رقم المجموعة فهو حقل أجنبي للربط بين الطالب و المجموعة التي ينتمي إليها.
- **Groupe** : جدول المجموعات و يضم الحقول التالية ( رقم المجموعة، اسمها) أما رقم التخصص NumFiliere فهو حقل أجنبي للربط بين المجموعة و التخصص



• **Filiere** : جدول التخصصات و يضم الحقول التالية (رقم التخصص، اسمه).

• **Absence** : وهو جدول الغياب و يضم الحقول الآتية (رقم الغياب و سنجدله تلقائي increment، و تاريخ الغياب، و عدد الساعات المتغيبه، و الملاحظات).

كخلاصة، قاعدة البيانات هذه ما هي إلا نموذج مبسط سنسعى من خلاله شرح المفاهيم المرومة.

و هذا هو كود إنشاء قاعدة البيانات :

```
create database Stagiaire
GO
use Stagiaire;

--table Filiere
create table Filiere(NumFiliere int,LibelleFiliere varchar(100),
Primary key(NumFiliere))

--table Groupe

create table Groupe (NumGroupe int,LibelleGroupe varchar(50),NumFiliere int
,Primary key(NumGroupe),Foreign key(NumFiliere) references Filiere)

--table stagiaire

create table Stagiaire(NumStagiaire int,Nom varchar(5),Prenom varchar(50),
Adresse varchar(250),DateNaissance datetime,Sexe varchar(50),NumGroupe
int,
primary key(NumStagiaire),Foreign key(NumGroupe) references Groupe)

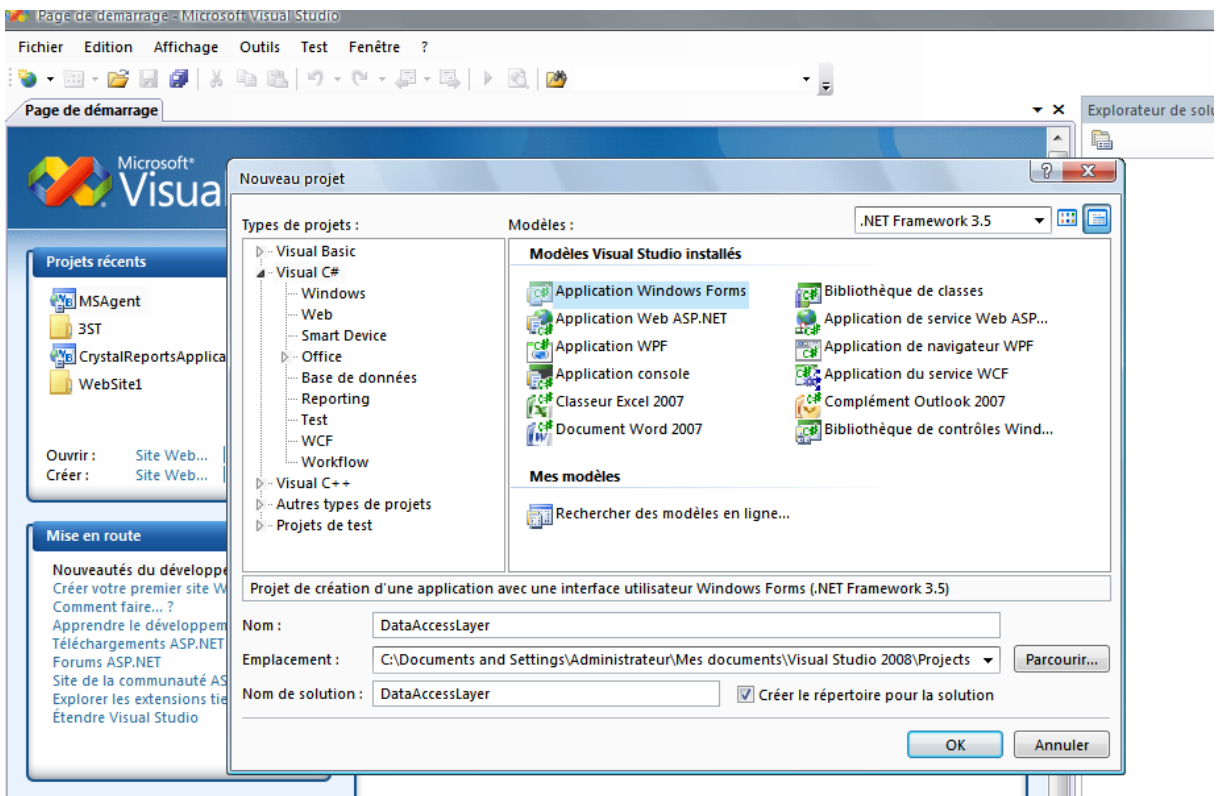
--table Absence

create table Absence(NumAbsence int,NumStagiaire int,DateAbsence Datetime,
NbrHeurs int,Observation nvarchar(250),Primary key(NumAbsence),
Foreign key(NumStagiaire) references Stagiaire)
```



الآن بعد أن أنشأت قاعدة البيانات إذهب إلى الفيچوال استوديو و قم بإنشاء مشروع جديد و ليكن

اسمه DataAccessLayer :



بعد ذلك سنقوم بإنشاء الفئة الرئيسية DataAccessLayer ولكن قبل ذلك تعال معي لتعرف عليها



الداتا أكسيس لاير هي قبل كل شيء فئة Class و من خلال اسمها نستنبط أنها تعني (طبقة الدخول إلى البيانات). بمعنى أنها القنطرة الرابطة بين البرنامج و بين قاعدة البيانات.

إذا لاحظت معي فقد ذكرنا هنا ثلاثة عناصر مهمة وهي :

- قاعدة البيانات
- الداتا أكسيس لاير
- البرنامج (أقصد بالبرنامج النوافذ و الفئات)

و هذا ما يسمى التطبيق متعدد الطبقات، أو ما يصطلح عليه ب MVC (Model,View,Controller)

الموديل هنا هو قاعدة البيانات، الفيو هو البرنامج، و الكونترولر هو الداتا أكسيس لاير.

و الغاية من هذه البنية هي تنظيم البرمجة و تقسيمها إلى أجزاء ليسهل التعامل مع كل جزء على حدى، و حتى نركز الشفرة الرئيسية في فئة واحدة تكون الجسر الفاصل بين البيانات و بين البرنامج، هذه الفئة التي ذكرنا بأنها الداتا أكسيس لاير.

من هنا نستنتج بأن الداتا أكسيس لاير ستضم دوال الربط بقاعدة البيانات و قراءة المعطيات و إرسالها.

سأفترض أنك على اطلاع بالإجراءات المخزنة stored procedure في sql server . إن كان الأمر كذلك فلنتابع إذن.

سننشئ فئة جديدة باسم الداتا أكسيس لاير .





و سنقوم باستيراد مجالات الأسماء التالية : System.Data.SqlClient, System.Data

وسنقوم بالاعلان عن متغير من نوع SqlConnection

بعد ذلك سننشئ مشيدا للفتة Constructor ومن خلاله سنحدد نص الكونيكشن.

أي سيكون الكود هكذا :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class DataAccessLayer
    {
        static SqlConnection SqlCon;

        public DataAccessLayer()
        {
            SqlCon = new SqlConnection("Server=TECHNI-AMECO-PC\\SQLEXPRESS;Database=Stagiaire;integrated security=true");
        }
    }
}
```

واحذر أن تقوم بنسخ الكود فعليك أن تغير نص الكونيكشن حسب نظامك.

بعد ذلك سننشئ دالتي ربط الإتصال و إغلاقه :



```
public void Connect()
{
    if (SqlCon.State != ConnectionState.Open)
        SqlCon.Open();
}

public void Disconnect()
{
    if (SqlCon.State != ConnectionState.Closed)
        SqlCon.Close();
}
```

و في الأخير سننشئ دالة لقراءة البيانات و دالة لإرسال البيانات :

```
public DataTable SelectData(String Stored, SqlParameter[]
Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length ; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlDataAdapter SqlDa = new SqlDataAdapter(SqlCmd);
    DataTable Dt = new DataTable();
    SqlDa.Fill(Dt);
    return Dt;
}
```

هذه الدالة تعيد لنا جدولاً DataTable لنتمكن من استغلاله في إظهار البيانات و اللعب بها كما نريد.



تأخذ الدالة متغيرين داخليين أولهما Stored وهو اسم الإجراء المخزن، و Param وهي البارامترات التي تنتظرها Stored Procedure.

و سنفهم هذا بالتفصيل فيما يلي.

ثم أعلننا عن متغير من نوع SqlCommand ليرسل البارامترات إلى الإجراء المخزن وذلك بتمريرها كلها عن طريق التكرار For.

بعد عملية ارسال البارامترات، ننشئ كائنا من SqlDataAdapter ليقوم بأخذ نتائج SqlCommand ووضعها في Datatable التي سنحصل نحن فيما بعد على البيانات من خلالها.

و هذه دالة إرسال البيانات :

```
public void ExecuteCommand(String Stored, SqlParameter[] Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlCmd.ExecuteNonQuery();
}
```

وهي مثل الدالة السابقة إلا أنها لا تقوم بقراءة البيانات و إنما بإرسالها.

أتمنى أن تكون قد تعرفت على الداتا أكسيس لاير ، إن كان العكس فلا تتردد في قراءة الفقرة جيدا فالأمر سهل جدا ويستلزم منك فقط شيئا من التركيز ، وهذا هو الكود كاملا:



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class DataAccessLayer
    {
        static SqlConnection SqlCon;

        public DataAccessLayer()
        {
            SqlCon = new SqlConnection("Server=TECHNI-AMECO-PC\\SQLEXPRESS;Database=Stagiaire;integrated security=true");
        }

        public void Connect()
        {
            if (SqlCon.State != ConnectionState.Open)
                SqlCon.Open();
        }

        public void Disconnect()
        {
            if (SqlCon.State != ConnectionState.Closed)
                SqlCon.Close();
        }

        public DataTable SelectData(String Stored, SqlParameter[] Param)
        {
            SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
            SqlCmd.CommandType = CommandType.StoredProcedure;
            for (int i = 0; i < Param.Length ; i++)
            {
                SqlCmd.Parameters.Add(Param[i]);
            }
            SqlDataAdapter SqlDa = new SqlDataAdapter(SqlCmd);
            DataTable Dt = new DataTable();
            SqlDa.Fill(Dt);
            return Dt;
        }
    }
}
```



```
public void ExecuteCommand(String Stored, SqlParameter[] Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlCmd.ExecuteNonQuery();
}
}
```

و الآن حان الوقت لنستغل الداتا أكسيس لاير، سنقوم بإنشاء فئة **Absence** تحتوي على دالتين : الأولى مشاهدة غياب الطالب من خلال رقمه، و الثانية إضافة غياب جديد للطالب:

هذه هي الدالة الأولى:

```
public DataTable SelectAbsence(int NumStagiaire)
{
    DataAccessLayer Dal = new DataAccessLayer();
    DataTable Dt = new DataTable();
    SqlParameter[] Param=new SqlParameter[0];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Dal.Connect();
        Dt=Dal.SelectData("SP_SelectAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
    return Dt;
}
```



بالنسبة لل SP\_SelectAbsence فهي إجراء مخزن Stored Procedure يقوم بجلب بيانات غياب الطالب من خلال رقمه و هذا هو كود الإجراء المخزن :

```
create proc SP_SelectAbsence
@NumStg int
as
select DateAbsence, NbrHeurs, Observation
from Absence where NumStagiaire=@NumStg
```

أعتقد أن الإجراء يشرح نفسه بنفسه، بحيث يتلقى رقم الطالب كبارامتر و يبحث عنه.

الدالة الثانية في الفئة Absence وهي دالة إضافة غياب جديد و لكن قبل ذلك سننشئ الإجراء المخزن الذي يقوم بعملية الإضافة:

```
create proc SP_AddAbsence
@NumStg int, @DateAbsence datetime, @NbrHeur int, @Observation nvarchar(250)
as
insert into absence (NumStagiaire, DateAbsence, NbrHeurs, Observation)
values (@NumStg, @DateAbsence, @NbrHeur, @Observation)
```

يعني قمنا بإضافة البارامترات التي يتلقاها الإجراء المخزن إلى الجدول Absence.

والآن هذه هي دالة الإضافة و التي تستعمل الداتا أكسيس لاير كوسيلة للوصول إلى الإجراء المخزن للكتابة في الجدول Absence.



```
public void AddAbsence(int NumStagiaire, DateTime DateAbsence, int
NbrHeurs, String Observation)
{
    DataAccessLayer Dal = new DataAccessLayer();
    SqlParameter[] Param = new SqlParameter[4];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Param[1] = new SqlParameter("DateAbsence",
SqlDbType.DateTime);
        Param[1].Value = DateAbsence;
        Param[2] = new SqlParameter("NbrHeur", SqlDbType.Int);
        Param[2].Value = NbrHeurs;
        Param[3] = new SqlParameter("Observation",
SqlDbType.NVarChar);
        Param[3].Value = Observation;
        Dal.Connect();
        Dal.ExecuteNonQuery("SP_AddAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
}
```

كما ترى قمنا بإرسال البارامترات إلى الإجراء المخزن عن طريق الوسيط الداتا أكسيس لاير. نرسل كل بارامتر بعدما نحدد اسمه ونوعه و يلزم أن يكونا مطابقين لمثليهما في الإجراء المخزن، و أدنى اختلاف قد يسبب خطأ.

ونرصد هنا شفرة الفئة Absence.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class Absence
    {
        public DataTable SelectAbsence(int NumStagiaire)
        {
            DataAccessLayer Dal = new DataAccessLayer();
            DataTable Dt = new DataTable();
            SqlParameter[] Param=new SqlParameter[1];
            try
            {
                Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
                Param[0].Value = NumStagiaire;
                Dal.Connect();
                Dt=Dal.SelectData("SP_SelectAbsence", Param);
            }
            catch
            {
                Dal.Disconnect();
            }
            return Dt;
        }
    }
}
```





```
public void AddAbsence(int NumStagiaire, DateTime DateAbsence, int
NbrHeurs, String Observation)
{
    DataAccessLayer Dal = new DataAccessLayer();
    SqlParameter[] Param = new SqlParameter[4];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Param[1] = new SqlParameter("DateAbsence",
SqlDbType.DateTime);
        Param[1].Value = DateAbsence;
        Param[2] = new SqlParameter("NbrHeur", SqlDbType.Int);
        Param[2].Value = NbrHeurs;
        Param[3] = new SqlParameter("Observation",
SqlDbType.NVarChar);
        Param[3].Value = Observation;
        Dal.Connect();
        Dal.ExecuteNonQuery("SP_AddAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
}
}
```

والآن حان موسم جني الثمار، كل ما قمنا بإنشائه من كود سنستعمله الآن ببساطة منقطة النظير، حيث في أسطر قليلة سنقوم بالكثير من المهام وهذه نقطة من أهم إيجابيات الداتا أكسيس لاير.



ليكن الفورم التالي هو فورم الغياب:

سنقوم بإنشاء شفرة إضافة غياب و ذلك بالضغط مرتين على الزر Ajouter.

```
Absence Ab = new Absence();  
Ab.AddAbsence(int.Parse(TxtID.Text),  
DateAbsence.Value,int.Parse(TxtNbr.Text), TxtObserv.Text);
```

لا تستغرب فهذا هو كود إضافة غياب جديد .



أما كود البحث فهو كما يلي:

```
Absence Ab = new Absence();  
DataTable Dt = new DataTable();  
Dt=Ab.SelectAbsence(int.Parse(TxtSearch.Text));  
this.DGV.DataSource = Dt;
```

أرأيت جمال الداتا أكسيس لاير !

ويمكن أن نعلن عن المتغير Ab مرة واحدة وهذا هو الكود الكامل :

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
namespace DataAccessLayer  
{  
    public partial class Form1 : Form  
    {  
        Absence Ab = new Absence();  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private void button2_Click(object sender, EventArgs e)  
        {  
            Ab.AddAbsence(int.Parse(TxtID.Text), DateAbsence.Value,  
int.Parse(TxtNbr.Text), TxtObserv.Text);  
        }  
    }  
}
```



```
private void button1_Click(object sender, EventArgs e)
{
    DataTable Dt = new DataTable();
    Dt=Ab.SelectAbsence (int.Parse (TxtSearch.Text));
    this.DGV.DataSource = Dt;
}
}
```

كنت أود أن أتعامل مع الداتا أكسيس لاير من خلال فئات جديدة لها علاقة بالتخصصات و المجموعات ولكنني الآن مقتنع تماما بأن نفس المبدأ سينطبق عليها لذا ارتأيت الإكتفاء بهذا فحسب.

لا تتشائم من مظهر الداتا أكسيس لاير فبشيء من التطبيق ستجد نفسك قد استسغتها كليا بل و أصبحت قادرا على أن تبدع فيها و بها.



إذا كان لديكم ملاحظات أو تساؤلات أو وجهت أخفءاء  
ففي الكتاب فلا تتردد أن تراسلني عبر

[Khalid\\_ESSAADANI@Hotmail.fr](mailto:Khalid_ESSAADANI@Hotmail.fr)