



## C# & Sql Server 2008 Tutorial (4)

မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမအားလုံး မင်္ဂလာပါဗျာ။ အခုဆိုရင် ကျနော်တို့ အနေနဲ့ C# & SQL Database ကိုလေ့လာခဲ့ကြတာ Project သို့ Login ဝင်ခြင်းအပိုင်း၊ Project Setting အား မွေနှောက်ခြင်းအပိုင်း အားလုံး ပြီးသွားခဲ့ပါပြီ။ ဒီအထဲတော့ အလွန်ကုန်ဆုံး နည်းလမ်းကို ရှာဖွေပြီး Command line လွန်လွန်လေး တွေ့နဲ့ ခရီးဆက်ခဲ့လို့ မိတ်ဆွေတို့အနေနဲ့လည်း လေ့ကျင့်နိုင်ကြလိမ့်မရှိလို့ ထင်မိပါတယ်။ ဒီအပတ် အပိုင်း(၄)မှာတော့ Login ဝင်ရောက်မှု Success ဖြစ်တဲ့အခါ မြင်တွေ့ရမရဲ့ Project ရဲ့ Main Page ကို UGI ချခြင်း၊ SQL Server အတွင်း Record များ သိမ်းဆည်းနိုင်ရန် Table တည်ဆောက်ခြင်းအပိုင်းလေးအထိ ဆွေးနွေးသွားမရဲ့လို့ မျှော်လင့်ထားပါတယ်။

အခုဆိုရင် တဖြည်းဖြည်းနဲ့ ကျနော်တို့ အမှန်တကယ် အလုပ်လုပ်မရဲ့ Project ကြီးရဲ့ Final Goal ကို စတင် ချဉ်းကပ်လို့ နေပါပြီ။ ဘာလို့လဲဆိုတော့ ကျနော်အနေနဲ့ ဒီ Tutorials တွေကို မှတ်ပုံတင်(သို့)နိုင်ငံသား စိစစ်ရေးကဒ် တစ်ခုပါ Data အချက်အလက်တွေကို Database အတွင်း သိမ်းဆည်းထားခြင်း၊ မှားယွင်းသည် များကို Update အဖြစ် ပြန်လည်ပြင်ဆင်ခြင်း၊ ထည့်သွင်းပြီး Record များကို Delete လုပ်ပစ်ခြင်း၊ Database အတွင်း သိမ်းဆည်းပြီး Record များထဲမှ မိမိသိချင်သော Data ကိုရှာဖွေခြင်း (သို့) Search Engine အပိုင်းလေး တွေ့ပြီးတဲ့အထိ ဆွေးနွေးသွားမရဲ့လို့ ရည်ရွယ်ထားလို့ပါ။ လေ့ကျင့်ခန်း အပိုင်းလေးတွေအနေနဲ့ အခုက အပိုင်း(၄)ဆိုတော့ Project အခုလုံး အပြီးဆိုရင် အပိုင်း(၆ သို့ ၇) လောက်အထိ ဖြစ်သွားလိမ့်မရှိလို့ ထင်မိပါတယ်။ ကြားထဲမှာ ဆွေးနွေးရင်းနဲ့လည်း စဉ်းစားမိတာလေးတွေကို ထည့်သွင်းဆွေးနွေး ပေးသွားမှာဖြစ်လို့ အပိုင်း နည်းနည်းရှည်ချင်လည်း ရှည်သွားနိုင်ပါတယ်။ အတတ်နိုင်ဆုံးတော့ အခြေခံလေးတွေအထိပါအောင် ဆွေးနွေး ပေးသွားပါမရဲ့။ ကျနော်ရဲ့ မကျွမ်းကျင်မှုကြောင့် မှားယွင်းတင်ပြခဲ့တာများ ရှိခဲ့ရင်လည်း နားလည်ပေးကြဖို့ အခုထဲက ကြိုတင်ပြောထားပါရစေဗျာ။ စကားတွေလည်း ရှည်နေပြီဆိုတော့ အလုပ်စကြဖို့...ဗျာ။

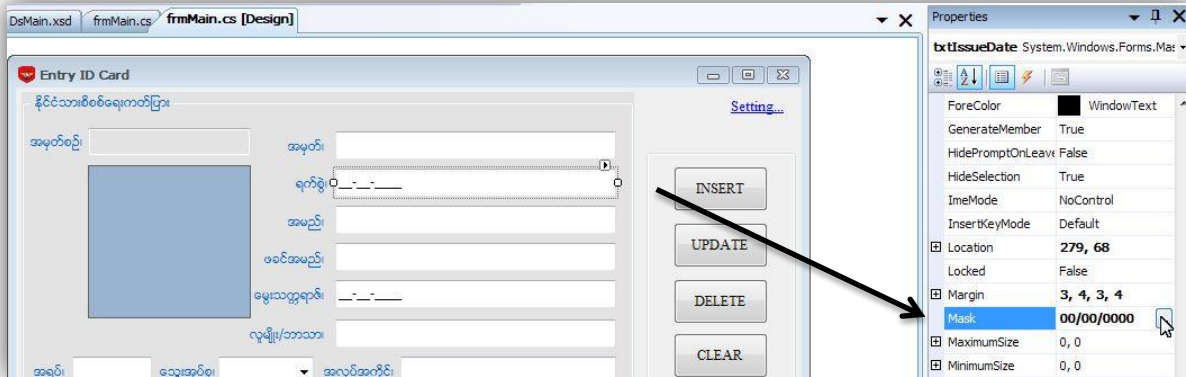
အခု ပထမဆုံးအနေနဲ့ Main Page ဖြစ်တဲ့ **frmMain** Form ရဲ့ UGI ကို အရင်ချပို့မရဲ့။ ကျနော်ကတော့ မှတ်ပုံတင် Record တွေကို Database အတွင်း သိမ်းဆည်းထားရမှာဖြစ်လို့ Fig.1 အတိုင်း ပုံစံချထားပါတယ်ဗျာ။

Fig.1

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...။

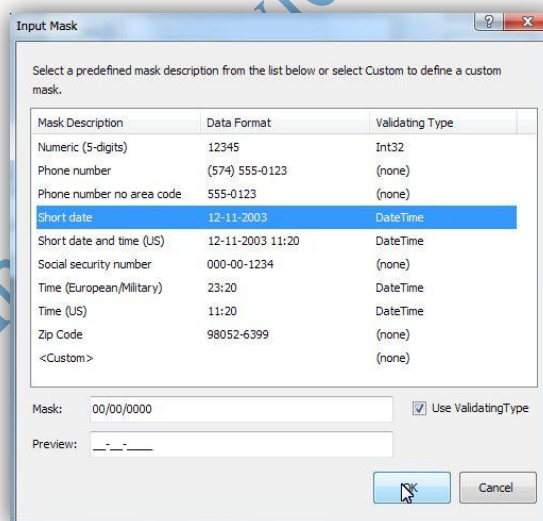


**frmMain** Form ကတော့ အပိုင်း(၁)မှာတည်းက တည်ဆောက်ခဲ့ပြီးဖြစ်လို့ အခုထပ်ပြီး Form တခု တည်ဆောက်တာကို မဆွေးနွေးပေးတော့ပါဘူး။ Tool လေးတွေရဲ့ အတိုင်ချမှု ပုံစံကိုတော့ **Fig.1** အတိုင်း ချထားလိုက်ကြပါဗျာ။ ဒီမှာ ကျနော်တို့ ထူးထူးခြားခြား MaskedTextBox(2)ခုနဲ့ ComboBox(1)ခုကို အသုံးပြုထားတာကို တွေ့ရမှာပါ။ ဒါကြောင့် ၎င်းတို့ရဲ့ အသုံးပြုနိုင်မှု Function လေးကို အနည်းငယ် ပြင်ယူကြည့်ကြရအောင်ဗျာ။ MaskedTextBox(2)ခုကတော့ Date တွေအတွက်သုံးထားတာပါ။ သူ့ရဲ့ Function ကိုပြင်ယူမရဲဆိုရင်တော့ ....



MaskedTextBox Fig.1

1. မိမိပြင်ယူချင်သော MaskedTextBox ကို Selected ပေးပြီး Properties Box → Mask ကို Click နှိပ်ပြီး ဝင်လိုက်ပါ။ **Input Mask** Box ပေါ်လာပါလိမ့်မည်။



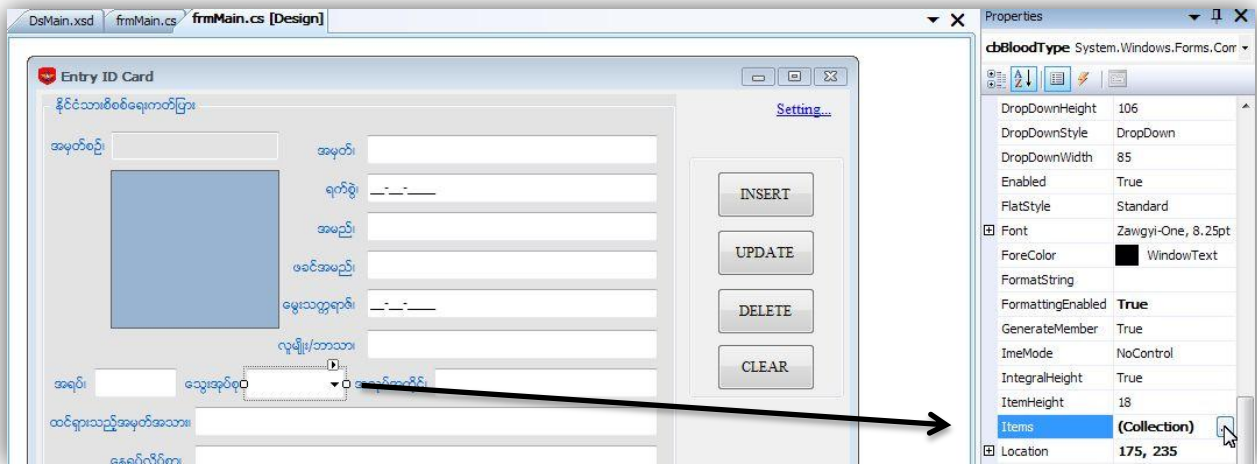
MaskedTextBox Fig.2

2. ၎င်းတွင် မိမိကြိုက်နှစ်သက်သော **Data Format** ကိုရွေးယူ အသုံးပြုနိုင်ပါတယ်။ ကျနော်ကတော့ အရိုးရှင်းဆုံးဖြစ်အောင် **Short date** Format ကို အသုံးပြုထားပါတယ်။ ဒီနည်းလမ်းအတိုင်းအတိုင်း **မွေးသက္ကရာဇ်** MaskedTextBox ကိုလည်း မိတ်ဆွေတို့ ကြိုက်နှစ်သက်သလို အသုံးပြုလိုက်ကြပေါ့ဗျာ။
3. ပေးထားတဲ့ ReadyMade တွေကိုမကြိုက်ဘူး။ မိမိစိတ်ကြိုက်ဖန်တီးချင်ရင်တော့ Custom ကိုရွေးချယ်ပြီး **Mask:** နေရာမှာ မိမိအသုံးပြုချင်သလို ဖန်တီး ယူနိုင်ပါတယ်။ **Preview:** ကတော့ ဘာလိုပေါ်လာမှာလဲဆိုတာကို ပြသပေးမဲ့နေရာပေါ့ဗျာ။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



အခုလို MaskedTextBox(2) ခုကို ကျနော်တို့ စိတ်ကြိုက်ပြင်ဆင်ယူပြီးသွားပြီဆိုရင်တော့ ကျန်ရှိနေသေးတဲ့ **ComboBox** လေးကို ဆက်လက်ပြီး ပြင်ယူကြည့်ကြမရဲ့။

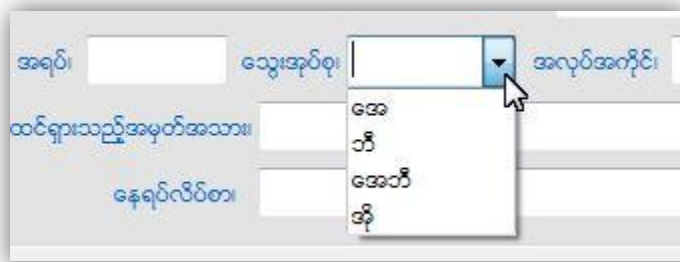


ComboBox Fig.1

1. ပုံမှန်ထားတဲ့အတိုင်း မိမိအသုံးပြုချင်သော ComboBox ကို Selected ပေးပြီး Properties → Items ကို Click ပေးပြီးဝင်လိုက်ပါ။ ဒါဆိုရင်တော့ ကျနော်တို့ ထည့်သွင်းချင်တဲ့ Items တွေကိုရေးသွင်းနိုင်ရန် အတွက် **String Collection Editor** Box ပေါ်လာပါလိမ့်မည်။
2. ၎င်း **String Collection Editor** Box ထဲတွင် မိမိအသုံးပြုချင်တဲ့ Items တွေကို တစ်ကြောင်းချင်းစီ ရိုက်သွင်းပြီး OK ပေးလိုက်ပါဗျာ။



ComboBox Fig.2



ComboBox Fig.3

3. ဒါဆိုရင်တော့ **ComboBox Fig.3** မှာပြထားသည့်အတိုင်း ကျနော်တို့ Project မှာ သွေးအုပ်စုကို ရွေးချယ်တဲ့အခါမှာ **Down Arrow Key** ကိုနှိပ်ပြီး မိမိကြိုက်နှစ်သက်ရာ သွေးအုပ်စုကို ရွေးချယ်ယူနိုင်ပါပြီဗျာ။ ဟုတ်ပြီဗျာ... ဒါဆိုရင်တော့ အခုဆက်လက်ပြီး Form UGI ရဲ့ Properties ကို ပြင်ယူကြည့်ကြအုံးစို့ဗျာ။

Control	Name	Text	TabIndex
TextBox1	txtNo	အမှတ်စဉ်	0
TextBox2	txtID	အမှတ်	1
MaskedTextBox1	txtIssueDate	ရက်စွဲ	2
TextBox3	txtName	အမည်	3

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



TextBox4	txtFatherName	ဖခင်အမည်	4
MaskedTextBox2	txtBirthDay	မွေးသက္ကရာဇ်	5
TextBox5	txtReligion	လူမျိုး/ဘာသာ	6
TextBox6	txtHeight	အရပ်	7
ComboBox1	cbBloodType	သွေးအုပ်စု	8
TextBox7	txtWork	အလုပ်အကိုင်	9
TextBox8	txtMark	ထင်ရှားသည့်အမှတ်အသား	10
TextBox9	txtAddress	နေရပ်လိပ်စာ	11
PictureBox1	picView	-	
DataGridView1	DgvRecord	-	

Control	Name	Text	TabIndex
Button1	btnInsert	INSERT	12
Button2	btnUpdate	UPDATE	13
Button3	btnDelete	DELETE	14
Button4	btnClear	CLEAR	15

အခုဆိုရင် **frmMain** Form ကြီးရဲ့ UGI ပုံစံချပြီးသလို ၎င်းအတွက် လိုအပ်တဲ့ Properties လေးတွေကိုလည်း ပြင်ယူပြီးခဲ့ပြီ ဖြစ်ပါတယ်။ အသေးစိတ်တော့ ရှင်းပြမနေတော့ပါဘူး။ **Name:** မှာပေးထားတဲ့ အမည်တွေကတော့ Coding ရေးတဲ့အခါမှာ ခေါ်သုံးရမှာဖြစ်လို့ နားလည်လွယ်ဖို့နဲ့ System ရဲ့ Keyword တွေနဲ့ မမှားမိဖို့လည်း သတိ ပြုသင့်ပါတယ်။ **Text:** ကစာသားတွေကတော့ Form မှာ ကျနော်တို့ မြင်တွေ့ကြရမရဲ့ စာသားတွေပဲဖြစ်ပါတယ်။ **TabIndex** ကတော့ ကျနော်တို့ Record တွေရှိကံသွင်းတဲ့အခါမှာ Text တစ်ခုကနေ Text တစ်ခုသို့ ကူးပြောင်းတဲ့ အခါမှာ Keyboard က Tab Key ကိုသုံးပြီး ပေးထားတဲ့ **No** အစဉ်လိုက် သွားစေချင်လို့ပဲဖြစ်ပါတယ်။ ရှေ့ကလေ့ ကျင့်ခန်းလေးတွေမှာ လေ့ကျင့်ခဲ့ပြီးဖြစ်လို့ အပေါ်က ဇယားတိုင်ကိုကြည့်လိုက်တာနဲ့ နားလည်သွားကြလိမ့်မရ် လို့ထင်ပါတယ်ဗျာ။ ရှေ့ဆက်ပြီးတော့ SQL Server 2008 ထဲမှာ Main Form အတွက် Table တစ်ခု တည်ဆောက် ကြည့်ကြရအောင်ဗျာ။

### Tbl\_Main Table တည်ဆောက်ခြင်း

SQL Server မှာ Table တွေကို တည်ဆောက်ရာမှာ ရှေ့ကလေ့ကျင့်ခန်းတွေမှာ Windows System နည်းနဲ့ တည်ဆောက်ခဲ့တာကို ဆွေးနွေးခဲ့ပြီးဖြစ်လို့ ဒီတစ်ခါတော့ အသွင်နည်းနည်းဆန်းသွားအောင် **New Query** တစ်ခုယူပြီး Command line ရေးကာ Table တစ်ခု တည်ဆောက်နိုင်တာလေးကို ဆွေးနွေးပေးသွားပါမရ်။ လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

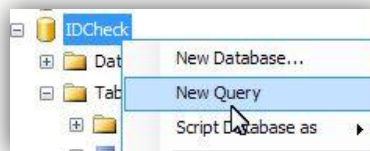


Fig.2

1. **Fig.2** မှာပြထားတဲ့အတိုင်း **IDCheck**(မိမိ Project ရဲ့ Database Name) → R-Click → New Query ကို Click ပေးလိုက်ပါဗျာ။ ဒါဆိုရင်တော့ ကျနော်တို့ Sql command line ရေးနိုင်ရန် Page တစ်ခုကျလာပါ လိမ့်မရ်။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



2. အောက်က command line လေးတွေကို ၎င်းပေါ်လာသော Page မှာ ရေးပြီး **Execute** Button လေးကို နှိပ်ပြီး Save လုပ်ပေးလိုက်ပါ။

Create Table Tbl\_Main

```
(
    NoId int identity primary key Not Null,
    No nvarchar(15) Null,
    ID nvarchar(20) Null,
    IssueDate nvarchar(15) Null,
    Name nvarchar(20) Null,
    FatherName nvarchar(20) Null,
    BirthDay nvarchar(15) Null,
    Religion nvarchar(20) Null,
    Hight nvarchar(15) Null,
    BloodType nvarchar(5) Null,
    Work nvarchar(50) Null,
    Mark nvarchar(50) Null,
    Adress nvarchar(100) Null,
    Picture varbinary(max) Null
)
```

3. ဒါဆိုရင်တော့ Fig. 3 ပါအတိုင်း ကျနော်တို့ **IDCheck** ဆိုတဲ့ Database ထဲမှာ **Tbl\_Main** ဆိုတဲ့ Table တစ်ခုရောက်ရှိလာ တာကိုတွေ့ရပါလိမ့်မယ်။

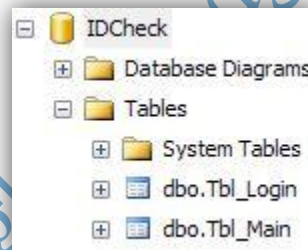


Fig.3

ဒီအခြေအနေမှာဆိုရင်ဖြင့် ကျနော်တို့အနေနဲ့ နိုင်ငံသားစိစစ်ရေးကဒ်(သို့)မှတ်ပုံတင် ပါ Data များကို သိမ်းဆည်း နိုင်ဖို့အတွက် **Tbl\_Main** Table က အသင့်ဖြစ်နေပါပြီ။ ဟုတ်ပြီ... နောက်တဆင့်အနေနဲ့ Record တွေ သိုလှောင်ထားနိုင်ဖို့ အသင့်ဖြစ်နေတဲ့ Tabl အလွတ်ကြီးထဲကို Data ကို ဘရ်လို့ ပုံစံနဲ့ သွားသိမ်းဆည်းထားမလဲ ကျနော်တို့ ထပ်စဉ်းစားကြည့်ကြရအောင်။ အခက်အခဲကြီးမရှိပါဘူး။ ရှေ့လေ့ကျင့်ခန်းတွေတုန်းကလိုပဲ **Stored Procedure** လေးတစ်ခုရေးပြီး သိမ်းထားလိုက်ယုံပါပဲ။

### Insert\_Main Stored Procedure တစ်ခုတည်ဆောက်ခြင်း

Stored Procedure တည်ဆောက်ပုံကိုတော့ ရှေ့လေ့ကျင့်ခန်းတွေမှာ ပုံစံနှစ်မျိုးနဲ့ ဆွေးနွေးခဲ့ပြီးဖြစ်တယ်။ ဒီလေ့ကျင့်ခန်းမှာတော့ Stored Procedure ကို C# → TablAdapter ထဲက မတည်ဆောက်တော့ပဲ၊ **SQL Server** ထဲက တည်ဆောက်သွားတာလေးနဲ့ ဆွေးနွေးပေးသွားပါမယ်။ မိတ်ဆွေတို့ကတော့ C# ထဲမှာပဲဖြစ်ဖြစ်၊ Sql Server ထဲကပဲဖြစ်ဖြစ် အားသန်ရာ ရေးသားကြပေမယ့် အလုပ်လုပ်ပုံချင်းကတော့ အတူတူပါပဲ။ ကဲ.. ရှေ့ဆက်ကြအုံးစို့။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



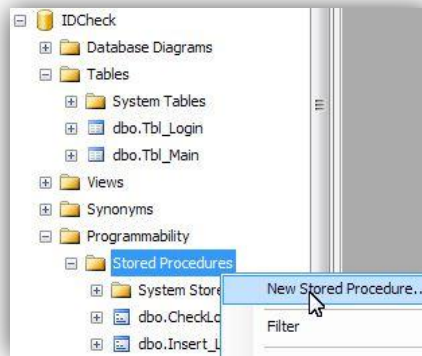


Fig.4

1. Fig.4 အတိုင်း IDCheck → Programmability → Stored Procedures → R-Click → New Stored Procedure.. → Click ပေးလိုက်ပါ။ ဒါဆိုရင်တော့ ကျနော်တို့ Stored Procedure အသစ်တစ်ခု ရေးသားနိုင်ဖို့အတွက် Page တစ်ခုကျလာပါလိမ့်မယ်။
2. အောက်က Sql Command line လေးအား ၎င်းပေါ်လာသော Page တွင် ရေးထည့်ပြီး Execute Button လေးအား နှိပ်ပြီး Save လုပ်လိုက်ပါ။

CREATE PROCEDURE [dbo].[Insert\_Main]

@No nvarchar(15),  
@ID nvarchar(20),  
@IsDate nvarchar(15),  
@Name nvarchar(20),  
@FName nvarchar(20),  
@BDay nvarchar(15),  
@Religion nvarchar(20),  
@High nvarchar(15),  
@BType nvarchar(5),  
@Works nvarchar(50),  
@Marks nvarchar(50),  
@Address nvarchar(100),  
@Pic varbinary(max)

AS  
BEGIN

Insert Into Tbl\_Main (No, ID, IssueDate, Name, FatherName, BirthDay, Religion, Hight, BloodType, Work, Mark,  
Adress, Picture )

Values(@No, @ID, @IsDate, @Name, @FName, @BDay, @Religion, @High, @BType, @Works, @Marks,  
@Address, @Pic)  
END

Table တည်ဆောက်စဉ်က ပေးခဲ့တဲ့ Column Name

အခက်ကြီးတွေမဟုတ်ပါဘူး။ ရှေ့မှာလည်း ပုံစံတူတွေ လေ့လာခဲ့ပြီ ဖြစ်ပါတယ်။ ကျနော်တို့ Tbl\_Main Table မှာ Column Field (14) ခုရှိပါတယ်။ NoId Column ကတော့ Record အရေအတွက် ဘယ်လောက်ရှိနေမလဲဆိုတာ သိနိုင်အောင် နံပါတ်တတ်ပေးထားတဲ့ သဘောပါပဲ။ ဒါကို ကျနော်တို့က Identity ပေးခဲ့လို့ ထည့်စဉ်းစားစရာ မလိုတော့ပါဘူး။ သူ့ဘာသာ Record တစ်ကြောင်း တိုးတိုင်း/လျှော့တိုင်း နံပါတ်စဉ် Auto အချိန်အပြောင်း လုပ်သွားပါလိမ့်မယ်။ ဒါဆိုရင် ကျနော်တို့ စဉ်းစားစရာ Column Field (13) ခုပဲ ကျန်တော့တာပေါ့ဗျာ။ အပေါ်က Command Line လေးတွေကိုကြည့်လိုက်ပါ။ ကျနော်တို့ Tbl\_Main Table ထဲကို အမှန်တကယ် Insert လုပ်ယူမရဲ့ Column Field(13)ခုကို သိုလှောင်ရာ Variable(13) ခုကို @ key လေးတွေခံပြီး Name(13)ခု ဦးစွာတည်

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...။



ဆောက်လိုက်ပါတယ်။ Table တည်ဆောက်တုန်းက ပေးခဲ့တဲ့ Column Name တွေနဲ့ တူစရာမလိုပါဘူး။ မိမိနားလည်လွန်သလို ပေးလိုက်နိုင်ပါတယ်။ နောက်က Data Type တွေနဲ့ သိုလှောင်ရာ Amount တွေတော့ တူညီစွာပေးထားသင့်ပါတယ်။

**Insert Into Tbl\_Main**(Insert လုပ်ယူမရဲ့ Column အားလုံး-Table တည်ဆောက်စဉ်က ပေးခဲ့သော Name နှင့် တူရမည်)  
**Values**(@ Insert လုပ်ရန် တည်ဆောက်ခဲ့ပြီးဖြစ်တဲ့ Variable အားလုံး)

ဒီ Command line ပုံစံကတော့ SQL မှာ Table တစ်ခုတည်းကို Record တွေသိမ်းဆည်းနိုင်ရန် အသုံးပြုတဲ့ Syntax ပုံစံလေးပါပဲ။ အသေမှတ်ထားလို့ရပါတယ်ဗျာ။ ဒီ Syntax လေးနဲ့ပဲ အပေါ်က လေ့ကျင့်ခန်း လေးမှာ Insert အတွက် Stored Procedure တစ်ခု ဖန်တီးသွားတာပါ။ အပေါ် command line တွေမှာပဲ ကျနော် များလေးတွေ ထိုးပြီး ရှင်းပြထားလို့ အလုပ်လုပ်သွားပုံကို နားလည်မရ်ထင်ပါတယ်။ အခုဆိုရင်တော့ ကျနော်တို့ IDCheck Database ထဲကို Insert\_Main ဆိုတဲ့ Stored Procedure တစ်ခုရောက်လာတာကိုတွေ့ရပါလိမ့်မယ်။ Fig.5 တွင် လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

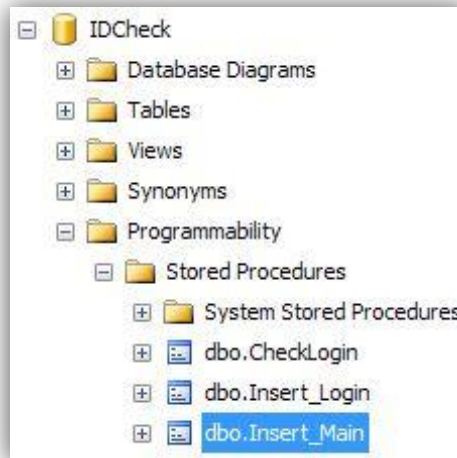


Fig.5

အခုဆိုရင် SQL Server ပေါ်မှာ frmMain Form အတွက် Tbl\_Main Table တစ်ခုလည်း တည်ဆောက်ခဲ့ကြပြီးပြီ ဒါအပြင် ၎င်း Table ထဲကို Record တွေ ထည့်သွင်းသိမ်းဆည်းထားနိုင်ဖို့အတွက် Insert\_Main ဆိုတဲ့ Stored Procedure တစ်ခုကိုလည်း ကျနော်တို့ တည်ဆောက်ခဲ့ကြပြီးပါပြီ။ ပြည့်စုံမှန်ကန်စွာ အလုပ်လုပ်ဖို့ ဒီ Insert လုပ်ငန်းစဉ်မှာ ဘာတွေလိုအပ်အုံးမလဲ frmMain Form ရဲ့ UGI ကိုကြည့်လိုက်ကြပါအုံးဗျာ။

Fig.6 ကိုကြည့်လိုက်ပါ။ ကျနော်တို့ Project ရဲ့ Insert လုပ်ငန်းစဉ်မှာ ဓါတ်ပုံတွေကို ထည့်သွင်းနိုင်ရန် အတွက် Picture Box တစ်ခုပါဝင်နေတာကို တွေ့ကြမှာပါ။ ဒါဆိုရင် ကျနော်တို့ စက်ထဲမှာရှိနေတဲ့ ဓါတ်ပုံတွေကို အခုမြင်နေရတဲ့ Picture Box ပေါ် ဘယ်လိုဆွဲတင်နိုင်မလဲ ဆိုတာကို ဦးစွာစဉ်းစားဖို့ လိုအပ်လာပြီပေါ့။ အခက်ခဲ ကြီးတော့ မဟုတ်ပါဘူးဗျာ။ အောက်မှာ လေ့လာကြည့်လိုက်ကြပါအုံး။

### Picture Box အတွင်းသို့ ဓါတ်ပုံဆွဲတင်ခြင်း

စက်ထဲမှာရှိနေတဲ့ File Type မတူတဲ့ ဓါတ်ပုံတွေကို UGI ရဲ့ Picture Box ပေါ်ကိုဆွဲတင်ဖို့က အခက်ကြီး မဟုတ်ပါဘူးဗျာ။ Window စနစ်က ခွင့်ပြုပေးထားတဲ့ Method လေးတွေကို အသုံးပြုလိုက်ယုံပါပဲ။ ကဲ... စလိုက်ကြအုံးစို့ဗျာ။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...

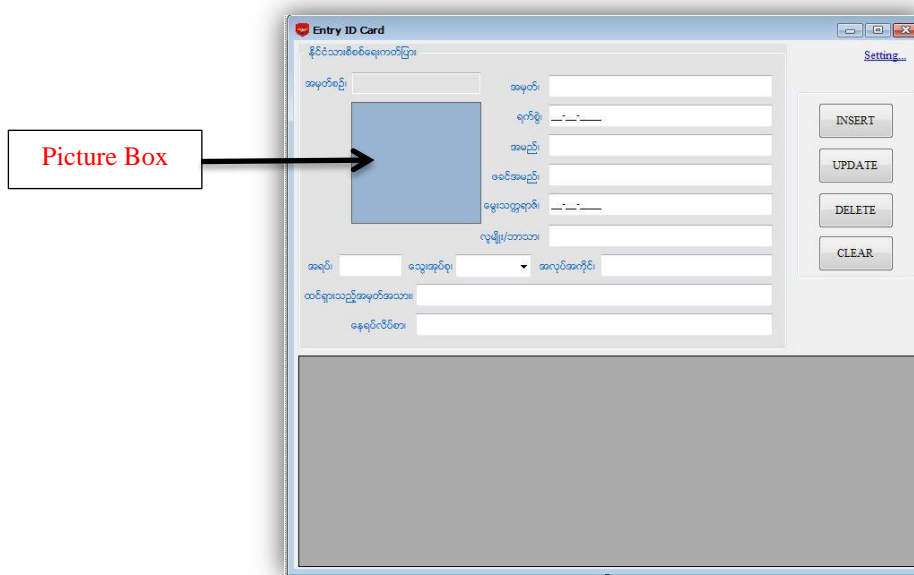


Fig.6

1. Toolbox → **OpenFileDialog** → တစ်ခုကို ကျနော်တို့ရဲ့ **frmMain** Form အတွင်းသို့ ဆွဲထည့်လိုက်ပါ။ ဒါဆိုရင်တော့ ကျနော်တို့ **frmMain** Form ရဲ့ အောက်ခြေလေးမှာ **OpenFileDialog** Box တစ်ခု ရောက်ရှိနေတာကိုတွေ့ရပါလိမ့်မရ။ ၎င်း Box ရဲ့ Properties ကိုလည်း ဆန္ဒရှိရင် ပြောင်းလဲယူနိုင်ပါတယ်။
2. ကျနော်ကတော့ နာမည်လေးတွေနဲ့ ပြန်ကြည့်ရလွယ်အောင် အခုလို ပြင်ယူလိုက်တယ်။

Control	Name	FileName
OpenFileDialog1	PicOpenFileDialog	PicOFD

ဟုတ်ပြီ။ ဒီအဆင့်ရောက်ပြီဆိုရင်တော့ ကျနော်တို့ Picture Box ပေါ်သို့ ဓါတ်ပုံတွေဆွဲတင်နိုင်ဖို့အတွက် Coding ရေးဖို့ အသင့်ဖြစ်နေပါပြီ။ အောက်မှာ ဆက်လေ့လာကြည့်လိုက်ပါအံ့ဗျာ။

1. **frmMain** Form → Picture Box → D-Click ပေးပြီး Coding ထဲသို့ ဝင်လိုက်ပါ။ ဒါဆိုရင်တော့ ကျနော်တို့ UGI ချတုန်းက **picView** လို့ နာမည်ပေးခဲ့တဲ့ Picture Box အတွင်းသို့ Coding ရေးရင် ရောက်ရှိသွားမှာပဲဖြစ်ပါတယ်။
2. အောက်က Coding လေးကို ရေးပြီး Save and Run လုပ်ပေးလိုက်ပါ။

```
private void picView_Click(object sender, EventArgs e)
{
    if (PicOpenFileDialog.ShowDialog() == DialogResult.OK)
    {
        picView.ImageLocation = Path.GetFullPath(PicOpenFileDialog.FileName);
    }
}
```

ဒါဆိုရင်တော့ ကျနော်တို့ PictureBox(picView) ကို Click ပေးကာ ပေါ်လာသော Open Dialog ထဲမှ မိမိကြိုက်နှစ်သက်ရာ ဓါတ်ပုံတစ်ပုံကို PictureBox အတွင်းသို့ ဆွဲတင်နိုင်ပါပြီဗျာ။ အလုပ်လုပ်သွားပုံလေးကို အနည်းငယ်ပြောပြလိုပါတယ်။ Coding ရဲ့ စတင်ချင်းမှာ **if( )** Statement တစ်ခုနဲ့ ကျနော်တို့ ရှေ့မှာ ခေါ်ယူခဲ့တဲ့ **OpenFileDialog** Box ကို Show လုပ်ခိုင်းပါတယ်။ ၎င်း Box ထဲမှာပဲ **DialogResult** အတွက် User က ဓါတ်ပုံတစ်ခုခုကို **OK(Open)** လုပ်လိုက်လျှင် **if( ) { ... }** တွန့်ကွင်းထဲက အလုပ်ကို ဆက်လုပ်မရဲဆိုပြီး





စစ်ထုတ်ထားပါတယ်။ User က Cancel ပေးလိုက်ရင်တော့ **if( )** Statement ကထွက်သွားမှာပဲဖြစ်ပါတယ်။ **Fig.7** ကိုကြည့်ရင်ရှင်းသွားပါလိမ့်မယ်။

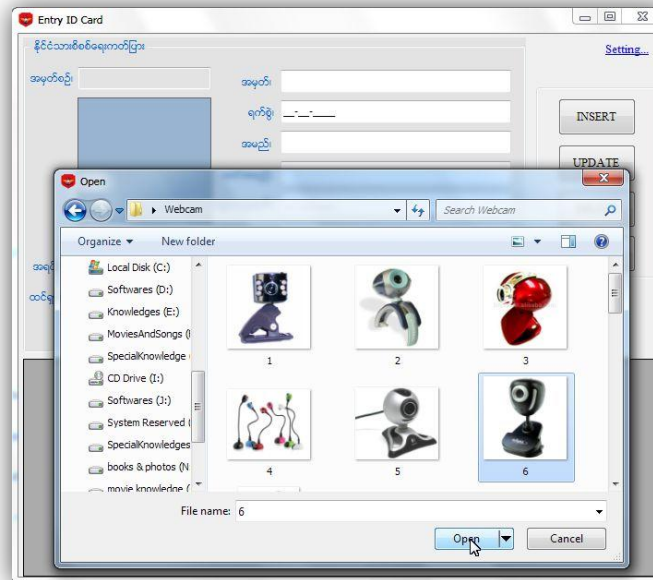


Fig.7

User က **Open(OK)** ပေးလိုက်ပြီဆိုရင်တော့ **if( )** Statement ကစတင် အလုပ်လုပ်ပါပြီ။ Coding လေးကို လေ့လာကြည့်လိုက်ပါ။ ကျနော်တို့ ရွေးချယ်ပေးလိုက်သော ဓါတ်ပုံရဲ့ ImageLocation (သို့) File Path လမ်းကြောင်းကြီးတစ်ခုလုံးကို **OpenFileDialog1(PicOpenFileDialog)** Box ကိုကြားခံထားပြီး ကျနော်တို့ တည်ဆောက်ထားသော Picture Box(picView) ပေါ်သို့ ဆွဲတင်လိုက်တာပဲဖြစ်ပါတယ်။ **Fig.8** ကိုကြည့်လိုက်ရင် တော့ နောက်ဆုံး Result ကို နားလည်သွားမှာပဲဖြစ်ပါတယ်။

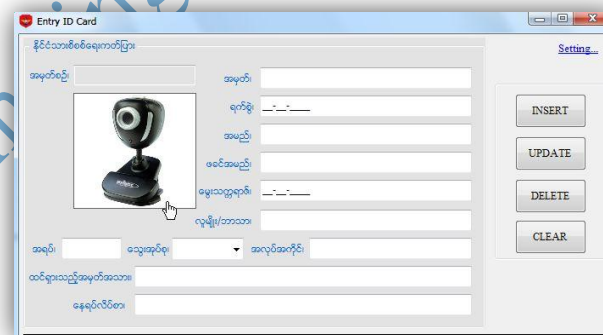


Fig.8

ဒီဓါတ်ပုံကိုမှ မကြိုက်သေးဘူးဆိုရင်လည်း Picture Box(picView) ကို ထပ်မံ Click ပေးပြီး မိမိစိတ်ကြိုက် ဓါတ်ပုံကို ရွေးယူနိုင်ပါတယ်။ Image File Type ကို အတိအကျ သတ်မှတ်ပေးထားခြင်း မရှိတဲ့အတွက် .jpeg, .png, .ico, ....., အစရှိတဲ့ မိမိ ကြိုက်နှစ်သက်ရာ Image Type တွေကို အသုံးပြုနိုင်ပါတယ်။

ဒီအပိုင်းမှာ မိတ်ဆွေတို့အနေနဲ့ အရေးအကြီးဆုံး သိထားသင့်တဲ့ အချက်တစ်ချက်ရှိနေပါသေးတယ်။ ဒါကတော့... C# ရဲ့ Library တစ်ခုကို ကြေငြာပေးဖို့ပဲ။

```
picView.ImageLocation = Path.GetFullPath(PicOpenFileDialog.FileName);
```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



ဒီ Command line မှာပါတဲ့ [Path.GetFullPath\( \)](#) ဆိုတဲ့ Key ကို C# ရဲ့ Library တစ်ခုဖြစ်တဲ့ Input/Output Header File ထဲကခေါ်သုံးထားတာပါ။ ဒါကြောင့် ဒီ Command Line ကို အလုပ်လုပ်စေချင်ရင် ကျနော်တို့ အနေနဲ့ အောက်က NameSpace(Header File) လေးကို ကြေငြာပေးရပါမယ်ဗျာ။

`using System.IO;`

ဒီ NameSpace လေးကို အထပ်မှာမကြေငြာခဲ့ဘူးဆိုရင်တော့ Picture Box အတွင်းသို့ Image တစ်ခုခုကို ဆွဲတင်တဲ့အခါ Error Message ပြပါလိမ့်မယ်ဗျာ။ နားလည်နိုင်ကြလိမ့်မယ်ထင်ပါတယ်။

**မှတ်ချက်။** ဒီလေ့ကျင့်ခန်းမှာ Image ဆွဲတင်ခြင်းအလုပ်ကို PictureBox အတွင်းမှ တိုက်ရိုက်မဟုတ်ဘဲ Button တစ်ခုခုအတွင်းမှလည်း Coding ရေးပြီးတော့ ခေါ်သုံးနိုင်ပါတယ်ဗျာ။

Sql Server မှာ Table တွေ၊ Stored Procedure တွေတည်ဆောက်ခဲ့ပြီးဖြစ်လို့ ၎င်း Sql ထဲက Table တွေ၊ Stored Procedure တွေကို C# မှာ ဘယ်လိုခေါ်သုံးနိုင်မလဲ ဆိုတာ ဆက်စဉ်းစားကြည့်ကြရအောင်။ လွန်ကူလွန်းပါတယ်။ DataSet တစ်ခုတည်ဆောက်မယ်၊ ၎င်း DataSet ထဲမှာ Table Adapter တစ်ခုတည်ဆောက်ပြီး ကျနော်တို့ Table ကို ဆွဲခေါ်အသုံးပြုယုံပဲပေါ့ဗျာ။ OK, ဒါဆို ရှေ့ဆက်ကြအုံးစို့။

### Tbl\_Main Table အတွက် DataSet တစ်ခုတည်ဆောက်ခြင်း

DataSet တည်ဆောက်တာနဲ့ ပတ်သတ်ပြီး ကျနော် အပိုင်း(၁)မှာ ရှင်းပြခဲ့ပြီးဖြစ်ပါတယ်။ ဒါကြောင့် Tbl\_Main Table ကို အရင်တည်ဆောက်ခဲ့ပြီးဖြစ်တဲ့ [DsLogin.xsd](#) ထဲကို ဆွဲတင်မသုံးဘဲ သီးခြား DataSet တစ်ခုတည်ဆောက် ရလည်းဆိုတာကို နားလည်နိုင်ကြလိမ့်မယ်ထင်ပါတယ်။ C# မှာ DataSet တစ်ခုတည်ဆောက်ခြင်းဟာလည်း အရမ်းကိုရိုးရှင်းပါတယ်။ အပိုင်း(၁)- [DsLogin.xsd](#) တည်ဆောက်ခဲ့တုန်းကလည်း လေ့လာခဲ့ပြီး ဖြစ်ကြပါတယ်ဗျာ။ ကဲ... စလိုက်ကြအုံးစို့။

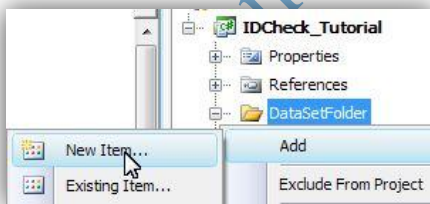


Fig.9

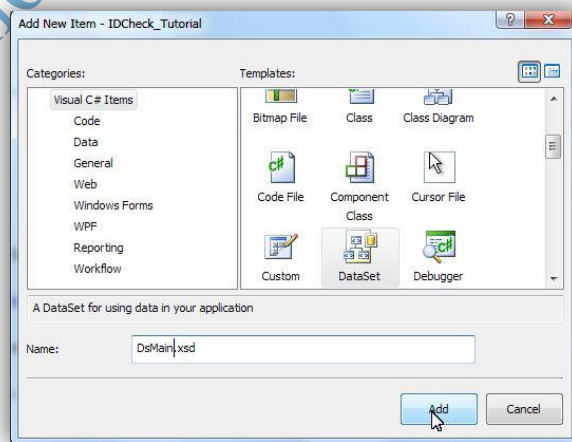


Fig.10

- Fig.9 အတိုင်းပါပဲဗျာ။ IDCheck\_Tutorial(Project Name) → DataSetFolder(ကျနော်တို့ Project ရဲ့ DataSet တွေ စုစည်းထားရန် တည်ဆောက်ခဲ့တဲ့ Folder) → R-Click → Add → New Item... → Click ပေးလိုက်ပါ။ ဒါဆိုရင်တော့ Fig.10 ကျလာတာကို တွေ့ရပါလိမ့်မယ်ဗျာ။
- ၎င်း Box ကို DataSet ကို Selected လုပ်ပေးကာ Name: တွင် မိမိနှစ်သက်ရာ နာမည်ပေးပြီး Add ပေးလိုက်ပါ။ ကျနော်ကတော့ Tbl\_Main ရဲ့ DataSet ဖြစ်လို့ DsMain.xsd လို့ ပေးထားပါတယ်။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



Fig.11

ဒါဆိုရင်တော့ Fig.11 အတိုင်း DataSet အလွတ်စာရွက်တစ်ခုကျလာပြီး ကျနော်တို့ DataSetFolder ထဲမှာ DsMain.xsd ဆိုတဲ့ DataSet တစ်ခုရောက်လာတာကိုတွေ့ရပါလိမ့်မယ်။ OK.. ဒါဆိုရင် ဒီ DataSet ထဲကို Tbl\_Main Table ကို ဆွဲတင်လိုရပြီပေါ့။ ဆက်လေ့လာကြည့်လိုက်အုံးစို့ဗျာ။

### Tbl\_MainTableAdapter တစ်ခုတည်ဆောက်ခြင်း

ဒီအပိုင်းလေးမှာတော့ ကျနော်တို့ တည်ဆောက်ပြီးဖြစ်တဲ့ DsMain.xsd DataSet ထဲကို Tbl\_Main ဆိုတဲ့ Table တစ်ခုကို ဆွဲတင်ကြည့်ကြမယ်။ လက်တွေ့ပြလိုက်ရင် အရမ်းရှိုးရှင်းပေမဲ့ စာနဲ့ရေးပြီး ရှင်းရတာ အရမ်းကို လက်ပေါက်ကပ်ပါတယ်။ ဒါပေမဲ့ ကျနော်အတတ်နိုင်ဆုံးနဲ့ အလွန်ရှာဖွေမှုကို ချဉ်းကပ်ပြီး ဆွေးနွေးသွား ပါမယ်ဗျာ။ အောက်က အဆင့်လေးတွေအတိုင်း လိုက်လုပ်ကြည့်လိုက်ကြပါ။

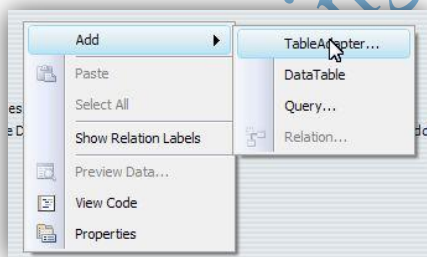


Fig.12



Fig.13

1. Fig.12 အတိုင်း DsMain.xsd DataSet Page အလွတ်ပေါ်မှာ R-Click → Add → TableAdapter... → Click ပေးလိုက်ပါ။ Fig.13 ပေါ်လာပါလိမ့်မည်။
2. Which data connection should your application use to connect to the database ? အောက်တွင် အပိုင်း(၁) Database Connection ချိတ်ဆက်ခြင်း(fig.12) မှာ ကျနော်တို့ ပေးခဲ့တဲ့ Connection Name ကို ရွေးပေးပြီး Next ပေးလိုက်ပါ (Window System ကတော့ ၎င်း Connection Name ကို Auto ရွေးပေးထားမည်ဖြစ်သည်။) Fig.14 ပေါ်လာမည်။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...

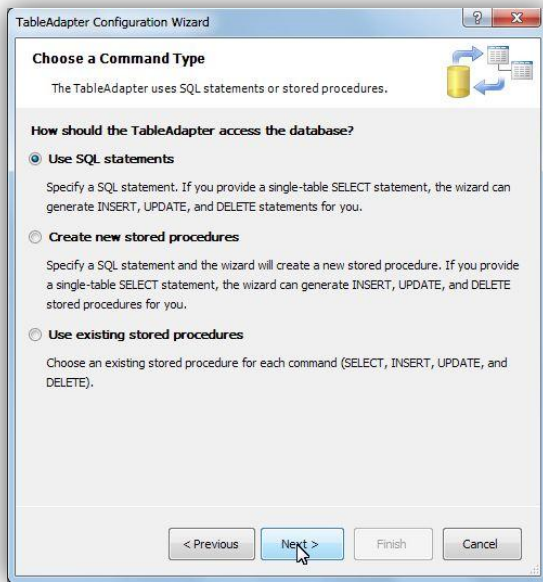


Fig.14



Fig.15

3. Use SQL statements ကို Selected ပေးပြီး Next သွားလိုက်ပါ။ Sql Command Line ရေးရန်အတွက် Fig.15 ပေးလာလိမ့်မည်။
4. ၎င်းတွင် **Select \* From Tbl\_Main** လို့ command line ရေးသွင်းပြီး Next ပေးလိုက်ပါ။ ဆိုလိုသည်က **Tbl\_Main** Table ထဲမှာရှိတဲ့ Column Field အားလုံးကို ဆွဲယူမရန်လို့ ပြောလိုက်ခြင်းပဲ ဖြစ်ပါတယ်။ Fig.16 ပေါ်လာပါလိမ့်မည်။

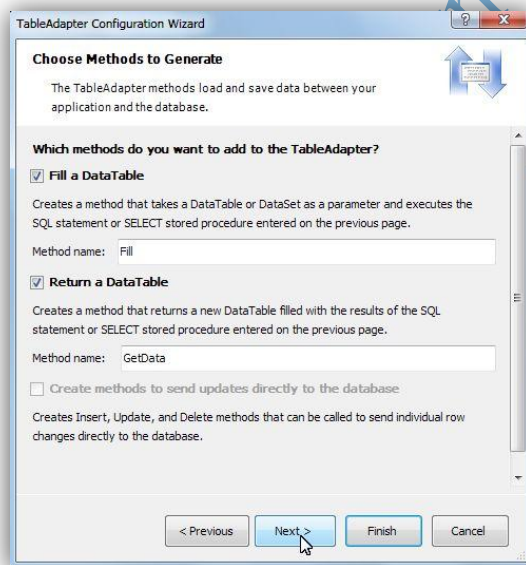


Fig.16

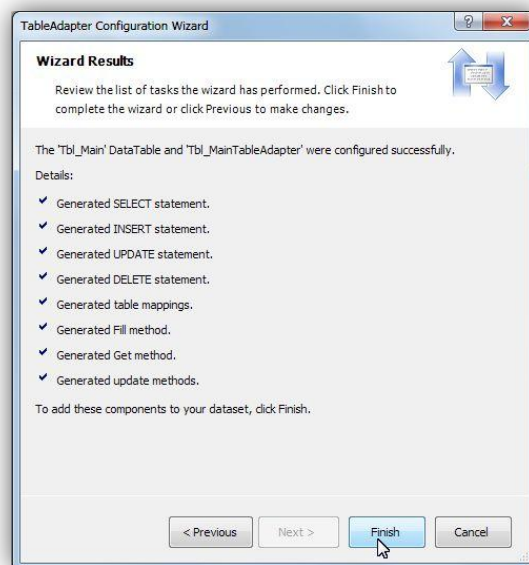


Fig.17

5. Fig.16 → Next → Fig.17 → Finish → Fig.18 ပေါ်လာပါလိမ့်မည်။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



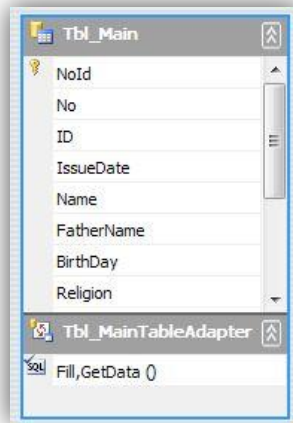


Fig.18

ဒါဆိုရင်တော့ Fig.18 မှာမြင်ရတဲ့အတိုင်း Tbl\_Main Table ရဲ့ TableAdapter တစ်ခုကို ရရှိပါပြီဗျာ။ ဆက်လက်ပြီး ကျနော်တို့အနေနဲ့ ဒီ TableAdapter ထဲသို့ ရှေ့မှာတည်ဆောက်ခဲ့ပြီးဖြစ်တဲ့ Insert\_Main Stored Procedure လေးကို ဘရ်လို လှမ်းဆွဲယူမလဲဆိုတာကို ဆက်လေ့လာကြည့်ကြမယ်။

### Tbl\_MainTableAdapter အတွင်းသို့ Insert\_Main Stored Procedure အား လှမ်းခေါ်ခြင်း

ကျနော်တို့အနေနဲ့ ဒီအပိုင်း(၄)မှာ Tbl\_Main Table ထဲကို မှတ်ပုံတင်(သို့) နိုင်ငံသားစိစစ်ရေးကဒ်ပါ Record တွေကို သိမ်းဆည်းသွားမှာဖြစ်ပါတယ်။ ဒီအတွက်ကြောင့်လဲ ရှေ့မှာ ကျနော်တို့ Insert\_Main ဆိုတဲ့ Stored Procedure လေးကို ကြိုတင် တည်ဆောက်ခဲ့ပြီးလည်းဖြစ်ပါတယ်။ ဒီလေ့ကျင့်ခန်းလေးမှာတော့ အဲဒီ SQL Server ထဲမှာ တည်ဆောက်ခဲ့တဲ့ Insert\_Main Stored Procedure လေးကို C# ရဲ့ TblAdapter ထဲက ဘရ်လို ခေါ်ယူနိုင်မလဲ ဆိုတာလေးကို ဆက်လက်ဆွေးနွေးပေးသွားမှာပဲဖြစ်ပါတယ်။ ဟုတ်ပြီ... ဆက်လေ့လာကြည့် လိုက်ပါအုံးဗျာ။

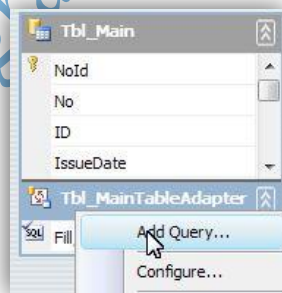


Fig.19

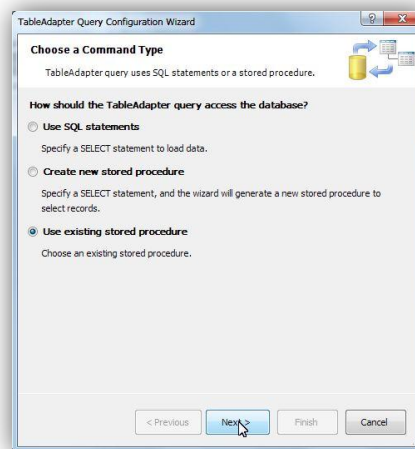


Fig.20

1. Fig.19 အတိုင်း Tbl\_MainTableAdapter → R-Click → Add Query.. → Click ပေးလိုက်ပါ။ Fig.20 ပေါ်လာလိမ့်မည်။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...





- ၎င်းတွင် **Use existing stored procedure** ကို **Selected** ပေးပြီး Next ပေးလိုက်ပါ။ ဆိုလိုသည်က Stored Procedure ကို ကျနော်တို့အနေနဲ့ သီးသန့်ရေးမနေတော့ပဲ ရေးပြီးသား/ရှိပြီးသား Stored Procedure ကို ခေါ်ယူမရဲ့လို့ ပြောလိုက်တာပါပဲဗျာ။ Fig.21 ပေါ်လာပါလိမ့်မည်။

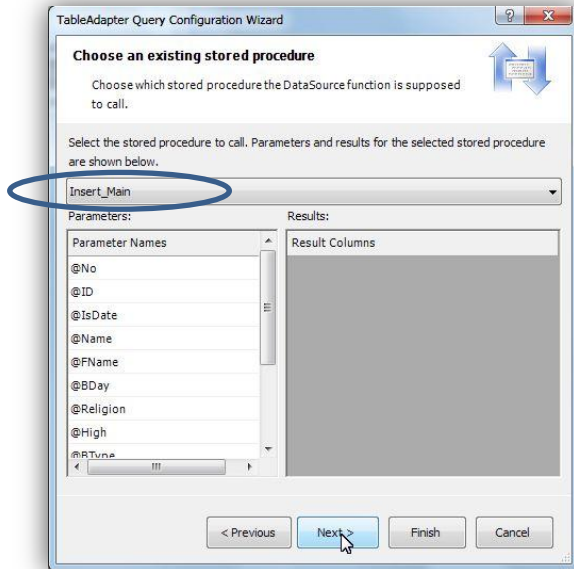


Fig.21

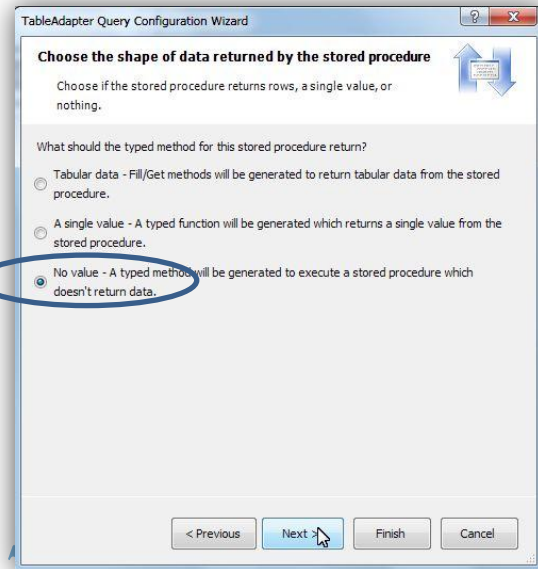


Fig.22

- Fig.21 ပါအတိုင်း ကျနော်တို့ အသုံးပြုချင်တဲ့ Stored Procedure ကိုရွေးပေးလိုက်ပြီး Next ပေးလိုက်ပါ။ အခုက Table **Tbl\_Main** ထဲကို Record တွေကို Insert လုပ်ပစ်မှာဖြစ်လို့ ကျနော်တို့ ကြိုရေးပြီးသား **Insert\_Main** Stored Procedure ကိုရွေးပေးလိုက်တာဖြစ်ပါတယ်။
- Fig.22 မှာတော့ **No value** ကို **Selected** ပေးပြီး Next ပေးလိုက်ပါ။

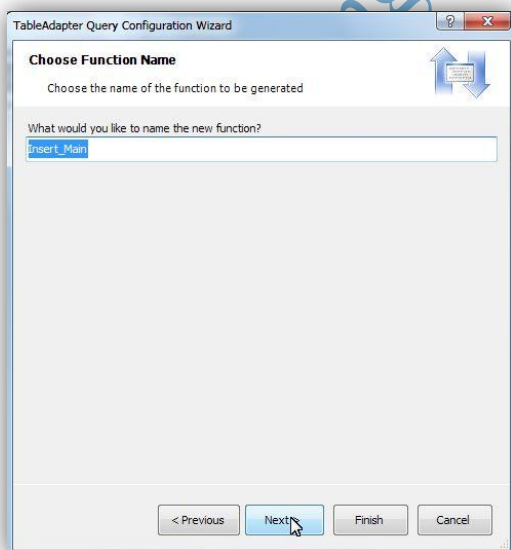


Fig.23

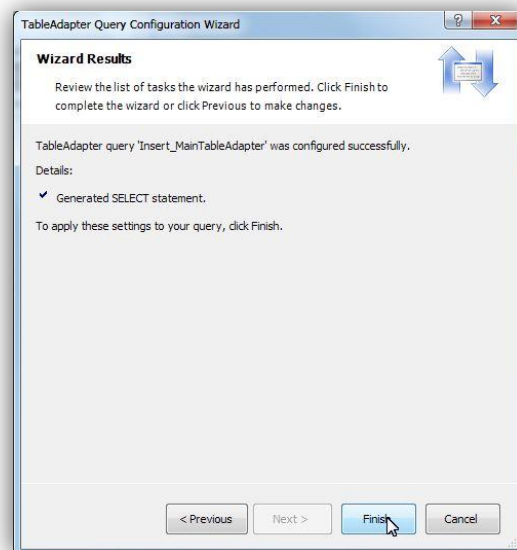


Fig.24

- Fig.23 → Next → Fig.24 → Finish ပေးလိုက်ပါ။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...

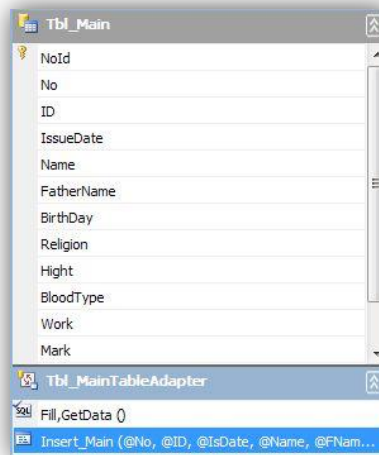


Fig.25

အခုဆိုရင် Fig.25 အတိုင်း Tbl\_MainTableAdapter ထဲကို **Insert\_Main** ဆိုတဲ့ Stored Procedure တစ်ခုရောက်ရှိလာတာ တွေ့ရပါလိမ့်မယ်။ ချိတ်ဆက်မှုတွေအားလုံး ပြည့်စုံသွားပြီဆိုတော့ ကျနော်တို့အနေနဲ့ ရှေ့ဆက်ပြီး C# မှာ Coding ရေးပြီး **Tbl\_Main** Table ထဲကို နိုင်ငံသားစိစစ်ရေးကဒ်(သို့)မှတ်ပုံတင်ပါ Record တွေ ဘရ်လို သိမ်းနိုင်မလဲဆိုတာ ဆက်လေ့လာကြည့်ကြအုံးစို့ဗျာ။

### C# တွင် Coding ရေးသားခြင်း

ဟုတ်ပြီဗျာ။ အခုဆိုရင်တော့ ကျနော်တို့ Project မှာ Insert လုပ်နိုင်ဖို့အတွက် စတင်ပြင်ဆင်လို့ရပါပြီ။ ရှေ့လေ့ကျင့်ခန်းတွေ တုန်းကလိုပဲ။ Namespace ကြေငြာတာလေးနဲ့ စလိုက်ကြရအောင်ဗျာ။

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

SQL Coding တွေကို Server အတွင်းကနေပြီး C# Coding မျက်နှာပြင်ပေါ်မှာ ခေါ်သုံးချင်တဲ့အခါ ဒီ Namespace ကိုကြေငြာရပါတယ်။ ကျနော်ဆွေးနွေးခဲ့တဲ့ လေ့ကျင့်ခန်းတွေမှာတော့ ဒီ Namespace က မကြေငြာမဖြစ် ကြေငြာရမရဲ့ Must Do ပေါ့။

ဆက်လက်ပြီးတော့ ကျနော်တို့ ချိတ်ဆက်ထားတဲ့ DataSet TableAdapter လမ်းကြောင်းကြီးကို coding ထဲမှာ ခေါ်သုံးနိုင်ဖို့အတွက် မိမိနှစ်သက်ရာ variable တစ်ခုအဖြစ် ပြောင်းလဲကြေငြာပြပါမရဗျာ။

```
DataSetFolder.DsMainTableAdapters.Tbl_MainTableAdapter tblMain = new DataSetFolder.DsMainTableAdapters.Tbl_MainTableAdapter();
```

ရှင်းရမရင်ဆိုရင်တော့ ကျနော်တို့ တည်ဆောက်ခဲ့ကြတဲ့ DataSetFolder အောက်ထဲက **Tbl\_MainTableAdapter** ကြီးတစ်ခုလုံးကို **tblMain** ဆိုတဲ့ Variable လေးထဲကို ပြောင်းလဲ သိမ်းယူလိုက်တာပဲဖြစ်ပါတယ်။ ဒါမှအောက်မှာ ကျနော်တို့ Coding ရေးတဲ့အခါ Database လမ်းကြောင်းကြီးတစ်ခုလုံးကို အထက်ကလို အဆင့်ဆင့် လိုက်ဝင်မနေရတော့ဘဲ **tblMain** ဆိုတဲ့ Variable လေးကိုခေါ်လိုက်တာနဲ့ **Tbl\_MainTableAdapter** လမ်းကြောင်းကို ခေါ်သုံးတရ်ဆိုတာ C# Coding ကသိရှိသွားမှာဖြစ်ပါတယ်။

ဆက်လက်ပြီး SQL Server ကြီးနဲ့ အဆက်အသွယ် ရရှိရန်အတွက် connection ချိတ်ဆက်ခြင်းကို ဆက်ကြည့်ကြရအောင်ဗျာ။

```
SqlConnection conn = new SqlConnection(Properties.Settings.Default.IDCheckConString);
conn.Open();
```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



အထက်က command line လိုပါပဲဗျာ။ SQL Server နဲ့ လှမ်းချိတ်ထားတဲ့ လမ်းကြောင်းကြီးတစ်ခုလုံးကို **conn** ဆိုတဲ့ variable ထဲကိုပြောင်းလဲသိမ်းယူလိုက်တာပဲဖြစ်ပါတယ်။ ဒီ Coding ကိုကြည့်မရဆိုရင် ကျနော်တို့ လေ့ကျင့်ခန်းအပိုင်း(၁) **Fig.11,12,13** မှာ ချိတ်ဆက်ခဲ့ဖူးကြတာကို ပြန်လည်သတိရကြမှာပါ။ အဲ့ဒီလမ်းကြောင်းကြီးကိုပဲ ဒီလေ့ကျင့်ခန်းမှာ ပြန်ခေါ်သုံးတာပဲဖြစ်ပါတယ်။ ဒါဆိုရင်တော့ C# Coding အပိုင်းမှာ Sql Server နဲ့ Connection ယူခြင်း ပြီးပြည့်စုံသွားပါပြီ။ ကျနော်တို့ ရှေ့ဆက်ပြီး **Tbl\_Main** Table ထဲကို နိုင်ငံသားစိစစ်ရေးကဒ်(သို့)မှတ်ပုံတင်ပါ Record များကိုသိမ်းဆည်းနိုင်ရန် အတွက် Coding ရေးယူရမရဲ့ အဆင့်ကိုရောက်ရှိလာပါပြီ။

ဒီမှာ ကျနော်တို့ အခက်အခဲအနည်းငယ်ရှိပါတယ်။ ဒါကတော့ ကျနော်တို့ Project အတွက် သိမ်းဆည်းရန် Record တွေထဲမှာ **Image** တွေ ပါလာလို့ပဲဖြစ်ပါတယ်။ ဒါကြောင့် Database ရဲ့ Table ထဲမှာ **Image** Type တွေကို ဘရလို သိမ်းဆည်းနိုင်မလဲ ကျနော်တို့ စဉ်းစားရပါပြီ။ ဘာမှမသိသေးဘဲ စတင်လေ့လာသူတွေအတွက် ဒီအဆင့်မှာ အခက်အခဲ အနည်းငယ်တော့ ရှိနိုင်ပါတယ်။ ကျနော်တုန်းကလည်း စဉ်းစားမရတော့လို့ အစ်ကိုကြီးတစ်ယောက်ကို မေးခဲ့ရပါတယ်။ ဖြေရှင်းနည်းကတော့ **Image** Type ကို **Byte** Type ပြောင်းလဲစေပြီး Table ထဲမှာ သိမ်းဆည်းလိုက်တာပါပဲ။ အခုတော့ ကျနော် မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမများအနေနဲ့ အရမ်းလွန်ကူသွားပါပြီ။ ကျနော်တုန်းက အချိန်ပေးခဲ့ရတာတွေကို မိတ်ဆွေတို့အဖို့ အခုတစ်ထိုင်တည်းနဲ့ ဖြေရှင်းသွားနိုင်ပါပြီ။ ကျနော်ပြောပြမရဲ့ အဆင့်လေးတွေအတိုင်း လုပ်ကြည့်လိုက်ပါဗျာ။ • ကျနော်လေ့ကျင့်ခန်းတွေကို မြန်မာလို အခုလိုရေးရခြင်းရဲ့ ရည်ရွယ်ချက်ကလည်း ဒါပါပဲ။ ကျနော် နားလည်သလောက်ကို စာဖတ်သူ မိတ်ဆွေတို့အဖို့ တစ်လုံး၊ တစ်ပါဒလောက် လွန်လွန်ကူကူ ရသွားစေချင်ခဲ့တာပါ။ ရှေ့ဆက်ကြစို့ဗျာ။

### Image Type မှ Byte Type ပြောင်းလဲခြင်း

ဒီမှာ Command Line လေးတွေကတော့ အနည်းငယ် ရှုပ်ထွေးပါတယ်။ ကျနော်ကိုယ်တိုင်လည်း အတိအကျကြီးတော့ နားမလည်သေးပါဘူး။ ဒါပေမဲ့ ဒီ code သုံးရင် ဒီအလုပ်လုပ်တစ်ခုဆိုတာကိုတော့ မိတ်ဆွေတို့ကို သိစေချင်တဲ့အတွက် ဒီလေ့ကျင့်ခန်းလေးကို ထည့်သွင်းပေးထားခြင်းပဲဖြစ်ပါတယ်။ ကျနော် အားနည်းချက်တွေအတွက် တောင်းပန်ပါတယ်ဗျာ။ အခုတော့ နားလည်သလိုလေး လေ့လာကြည့်လိုက်ကြရအောင်။ ဖြေရှင်းနည်းကတော့ ကျနော်အပေါ်မှာ ပြောခဲ့သလိုပါပဲ... **Image** Type ကို **Byte** Type ပြောင်းပြီး Database ထဲမှာ သိမ်းဆည်း လိုက်တာပါ။ ဒါကြောင့် ကျနော်တို့ **ConvertImageToByte** လို့အမည်ပေးထားတဲ့ Byte Function လေးတစ်ခု ရေးယူလိုက်ကြရအောင်။ **frmMain.cs** Form ထဲမှာ အောက်ပါ **Byte[ ]** Function လေးကို ရေးယူလိုက်ပါဗျာ။

```
byte[ ] ConvertImageToByte(string filesource)
{
    FileInfo info = new FileInfo(filesource);
    FileStream Fs = new FileStream(filesource, FileMode.Open, FileAccess.Read);
    BinaryReader Br = new BinaryReader(Fs);
    byte[ ] returnByte = Br.ReadBytes((int)Fs.Length);
    return returnByte;
}
```

နားလည်သလောက် အနည်းငယ်ရှင်းရမရန်ဆိုရင်တော့ ကျနော်တို့ **Image** မှ **Byte** Type ပြောင်းလဲပေးမရဲ့ **Byte** Function ကို **ConvertImageToByte** ဆိုပြီး အမည်ပေးထားပါတယ်။ ပထမ Command line မှာတော့ ကျနော်တို့ **PictureBox(picView)** မှာ ရွေးချယ်ပေးလိုက်တဲ့ **Image** ဖိုင်ရဲ့ **Info** ကို လှမ်းဖတ်ပြီး **info** ဆိုတဲ့ Variable ထဲကို



သိမ်းဆည်းလိုက်ပါတယ်။ ဒုတိယ Command line မှာတော့ ကျနော်တို့ လှမ်းဖတ်ရရှိတဲ့ Image File ရဲ့ လမ်းကြောင်းကြီး တစ်ခုလုံးကို ဆွဲယူပြီး Fs ဆိုတဲ့ Variable ထဲမှာ သိမ်းဆည်းလိုက်ပါတယ်။ တတိယ line ကတော့ ကျနော်တို့ ရရှိတဲ့ Image ကို Binary အဖြစ်ပြောင်းလဲဖို့အတွက် BinaryReader ဆိုတဲ့ Key ကို အသုံးပြုပြီး Br ဆိုတဲ့ Variable ထဲကိုထည့်လိုက်ပါတယ်။ နောက်ဆုံး Line ကတော့ ကျနော်တို့ ပြောင်းလဲ ယူလိုက်တဲ့ Binary No တွေကို Result အဖြစ် Database ထဲမှာ သိမ်းဆည်းနိုင်ဖို့အတွက် returnByte ဆိုတဲ့ Variable ထဲမှာ ပြောင်းလဲ သိမ်းဆည်းပြီး ၎င်း returnByte ကို Result အနေနဲ့ Return ပြန်ပေးထားတာပဲဖြစ်ပါတယ်။ ဒါဆိုရင် တော့ ကျနော်တို့ PictureBox(picView) မှရွေးချယ်ပေးလိုက်တဲ့ Image တွေကို Byte Type ပြောင်းပြီး Database ထဲမှာ သိမ်းဆည်းချင်တဲ့အခါ ဒီ ConvertImageToByte( ) Function လေးနဲ့ ပြောင်းလဲအသုံးပြုနိုင်ပါပြီဗျာ။ ဟုတ်ပြီဗျာ... ဒါဆိုရင် ကျနော်တို့ Database ရဲ့ Tbl\_Main Table ထဲကို နိုင်ငံသားစိစစ်ရေးကဒ်(သို့)မှတ်ပုံတင်ပါ Record များကို သိမ်းဆည်းနိုင်ရန်အတွက် INSERT Button ထဲမှာ Coding ဝင်ရေးကြည့်ကြရအောင်ဗျာ။

### INSERT Record to Tbl\_Main Table

ကျနော်တို့ ရေးနေကြပုံစံအတိုင်းပါပဲ။ frmMain.cs Form မှ INSERT Button ကို D-Click ပေးပြီး Coding Page အတွင်းသို့ ဝင်လိုက်ပါ။ အောက်ပါ Coding လေးကို INSERT button ထဲမှာရေးပြီး Run ကြည့် လိုက်ပါဗျာ။

```
private void btnInsert_Click(object sender, EventArgs e)
{
    try
    {
        if(txtNo.Text!="" & txtID.Text!="" & txtIssueDate.Text!="" & txtFatherName.Text!="" & txtName.Text!="" &
        txtBirthDay.Text!="" & txtReligion.Text!="" & txtHeight.Text!="" & cbBloodType.Text!="" & txtWork.Text!="" &
        txtMark.Text!="" & txtAddress.Text!="" & picView.ImageLocation!="")
        {
            tblMain.Insert_Main(txtNo.Text, txtID.Text, txtIssueDate.Text, txtFatherName.Text, txtName.Text, txtBirthDay.Text,
            txtReligion.Text, txtHeight.Text, cbBloodType.Text.ToString(), txtWork.Text, txtMark.Text, txtAddress.Text,
            ConvertImageToByte(picView.ImageLocation.ToString()));
            MessageBox.Show("သိမ်းဆည်းပြီးပါပြီ","Insert");
        }
        else
        {
            MessageBox.Show("ကျေးဇူးပြုပြီး Data များကို စုံလင်ပြည့်စုံစွာဖြည့်သွင်းပါ။","Insert");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

ဒီလေ့ကျင့်ခန်းလေးမှာ try{..}catch{..} ဆိုတဲ့ Command line လေးကို ကျနော်ထည့်သွင်းအသုံးပြုပြ ထားပါတယ်။ ၎င်းရဲ့အလုပ်လုပ်ပုံကတော့ သူ့ရဲ့ coding ကိုစတာနဲ့ try{...} အလုပ်ကို စလုပ်ပါတယ်။ ၎င်း try{...} ထဲမှာ ပြဿနာတစ်ခုခုတတ်ပြီဆိုရင်တော့ ဒါဟာ ဘာ Error အမှားဖြစ်တစ်ခုဆိုတာကို catch{...} ကလှမ်းဖမ်းပြီး Message Box အနေနဲ့ ကျနော်တို့ကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။ Error Message ကိုဖတ်ပြီး ကျနော်တို့ ဘာကြောင့် မှားတစ်ခုဆိုတာကို သိနိုင်ပြီး အမှားကို ပြုပြင်နိုင်တာပေါ့ဗျာ။ ဒီလောက်ဆိုရင်တော့ Try{...}Catch{...} ရဲ့ အလုပ်လုပ်ပုံကို နားလည်နိုင်ကြပြီလို့ ထင်ပါတယ်။ Command line လေးတွေကို ဆက်ရှင်းလိုက်ကြရအောင်။





စစချင်းမှာ if( ) Statement တစ်ခုအသုံးပြုပြီးတော့ ကျနော်တို့ Table ထဲကို ရိုက်သွင်းသိမ်းဆည်းမရ် Record (or) Column Filed(13) ခုဟာ !="" ဖြစ်နေမရ် (သို့) Record အားလုံးရိုက်ထည့်ထားမရ်ဆိုရင် if Statement ရဲ့ {...} ကို အလုပ်လုပ်ပါလို့ စစ်ထုတ်ထားပါတရ်။ တကရ်လို့ !="" ဖြစ်မနေဘူး။ တနည်းအားဖြင့် Record တစ်ခုရ ရိုက်သွင်းဖို့ လိုအပ်နေမရ်ဆိုရင်တော့ else {...} Statement ကို ဆင်းအလုပ်လုပ်စေပြီး "ကျေးဇူးပြုပြီး Data များကို စုံလင်ပြည့်စုံစွာဖြည့်သွင်းပါ။" ဆိုတဲ့ Message Box ကို ဖော်ပြပေးမှာဖြစ်ပါတရ်။

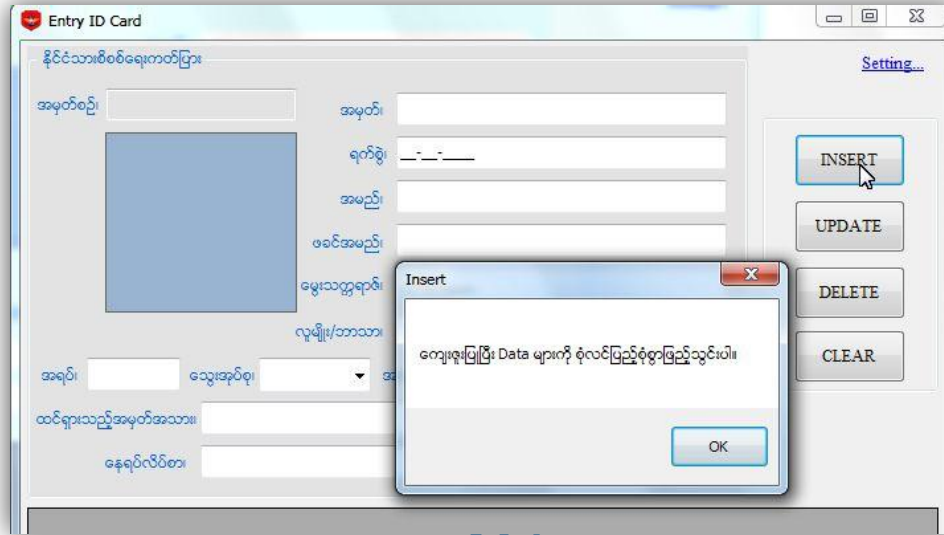


Fig.26

Fig.26 ကိုကြည့်လိုက်ပါဗျာ။ ဒီလောက်ဆို ရင်ဖြစ် if( ) {...} else {...} Statement အလုပ်လုပ်ပုံကို နားလည်သွားပြီလို့ထင်ပါတရ်။ နောက် Command line လေးကို ဆက်လိုက်ကြအုံးစို့။

If( ) Statemet ကတော့ !="" ဖြစ်နေပြီဆိုရင်တော့ if{...} ထဲက အလုပ်ကိုဆက်လုပ်ပြီပေါ့။ ဒီမှာပဲ အခက်အခဲကြီး မရှိပါဘူး။ ရှေ့လေ့ကျင့်ခန်းတွေမှာ ကျနော်တို့ လေ့လာခဲ့ကြပြီးဖြစ်ပါတရ်။

```
}
tblMain.Insert_Main(txtNo.Text,txtID.Text,txtIssueDate.Text,txtFatherName.Text,txtName.Text,txtBirthDay.Text,
txtReligion.Text,txtHeight.Text,cbBloodType.Text.ToString(),txtWork.Text,txtMark.Text,txtAddress.Text,
ConvertImageToByte(picView.ImageLocation.ToString()));
MessageBox.Show("သိမ်းဆည်းပြီးပါပြီ","Insert");
}
```

ဦးစွာ Tbl\_MainTableAdapter နဲ့ Connection ချိတ်မိနိုင်ဖို့အတွက် tblMain လို့ ကျနော်တို့ ကြေငြာခဲ့တဲ့ Variable ကို လှမ်းခေါ်လိုက်ပါတရ်။ ပြီးမှ ၎င်း TableAdapter ထဲက ကျနော်တို့ အပေါ် Tbl\_MainTableAdapter အတွင်းသို့ Insert\_Main Stored Procedure အား လှမ်းခေါ်ခြင်း လေ့ကျင့်ခန်းမှာ ဆွဲခေါ်ခဲ့ပြီးဖြင့်တဲ့ Insert\_Main Stored Procedure ကိုလှမ်းလိုက်ပါတရ်။ ၎င်းနောက် ကျနော်တို့ Keyboard ကရိုက်ထည့်လိုက်တဲ့ Record(13) ခုကို Insert\_Main Stored Procedure ရဲ့ @ နဲ့စခဲ့တဲ့ Variable(13)ခုနဲ့ ညီပေး လိုက်ပြီး Insert အလုပ်ကို လုပ်ပေးသွားတာဖြစ်ပါတရ်။ ဒါကြောင့် ကျနော်တို့ frmMain.cs Form မှာ ရိုက်သွင်း လိုက်သမျှသော Data တွေဟာ SQL Server ထဲက Tbl\_Main Table ထဲကို သွားရောက်သိမ်းဆည်း ပေးနိုင်ခဲ့တာပဲ ဖြစ်ပါတရ်။

ဒီ Command မှာ တစ်ခုထူးခြားသွားတာကတော့ ကျနော်တို့ Image ကို သိမ်းဆည်းတဲ့အခါမှာ ConvertImageToByte(picView.ImageLocation.ToString()) ဆိုပြီး ကျနော်တို့ အပေါ်မှာ ကြိုတင်တည်ဆောက်ခဲ့တဲ့

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...





**ConvertImageToByte( )** Function လေးကိုအသုံးပြုပြီး **Image** Type ကို **Binary** Type အဖြစ်ပြောင်းလဲသိမ်းဆည်းလိုက်ခြင်းပဲဖြစ်ပါတယ်။ Function တစ်ခုခေါ်သုံးခြင်းကိုတော့ နားလည်နိုင်ကြလိမ့်မယ် ထင်ပါတယ်။ Screen Shoot လေးကိုကြည့်လိုက်ကြပါအုံးဗျာ။

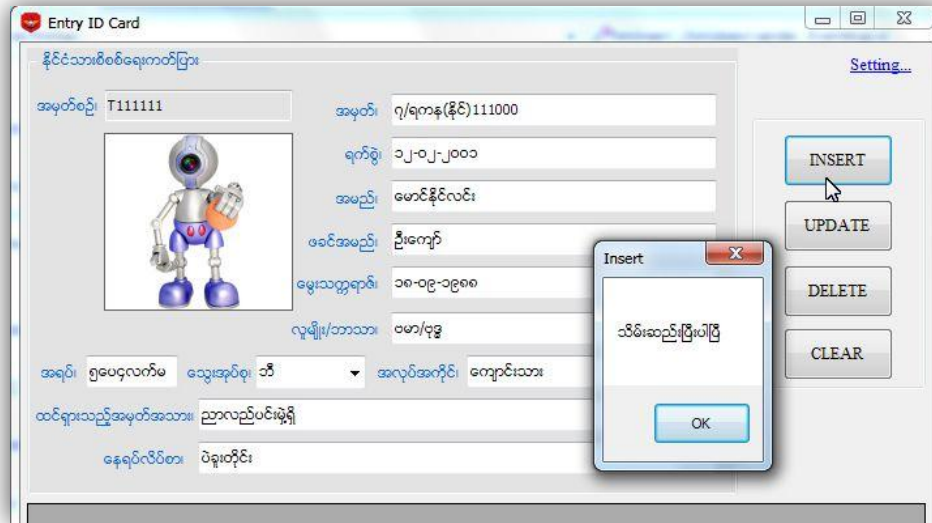


Fig.27

N...	No	ID	IssueDate	Name	FatherName	BirthDay	Religion	Hight	BloodT...	Work
1	T111111	၇/ရကန(နိုင်)111000	၁၂-၀၂-၂၀၀၁	ဦးကျော်	မောင်နိုင်လင်း	၁၈-၀၉-၁၉၈၈	ဗမာ/ဗုဒ္ဓ	၅ပေ၄လက်မ	ဘီ	ကျောင်းသား
N...	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig.28(a)

Name	FatherName	BirthDay	Religion	Hight	Bloo...	Work	Mark	Adress	Picture
ဦးကျော်	မောင်နိုင်လင်း	၁၈-၀၉-၁၉၈၈	ဗမာ/ဗုဒ္ဓ	၅ပေ၄လက်မ	ဘီ	ကျောင်းသား	ညာလည်ပင်းဖွဲ့ရုံ	ပဲခူးတိုင်း	<Binary data>
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig.28(b)

Fig.27 ကတော့ Record အားလုံးရိုက်သွင်းပြီးလို့ **INSERT** လုပ်ငန်းအောင်မြင်သွားတဲ့ အခြေအနေပဲဖြစ်ပါတယ်။ Fig.28(a,b) ကတော့ ကျနော်တို့ SQL Server ရဲ့ **Tbl\_Main** Table ထဲမှာ ကျနော်တို့ ရိုက်သွင်းသိမ်းဆည်းလိုက်တဲ့ Data တွေကို မိတ်ဆွေကို တစ်ခါတည်း ရှင်းရှင်းလင်းလင်းမြင်သွားအောင် ဖော်ပြပေးခြင်းဖြစ်ပါတယ်။ Table ရဲ့ **Picture** Column Field နေရာကိုကြည့်လိုက်ပါဗျာ။ ကျနော်တို့ သိမ်းဆည်းလိုက်တဲ့ Image ကို **<Binary data>** အဖြစ်ပဲ ဂုဏ်ပြီး သိမ်းဆည်းပေးထားတာ တွေ့ရပါလိမ့်မယ်။ ဒီလောက်ဆိုရင်တော့ **ConvertImageToByte( )** Function လေးရေးပြီး Image ကို Byte ပြောင်းလဲသိမ်းဆည်းရခြင်းရဲ့ အကျိုးကို သိသွားပြီလို့ ထင်ပါတယ်ဗျာ။

မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမများအနေနဲ့ သေချာနားလည်သွားအောင် **frmMain.cs** ရဲ့ Coding အပြည့်အစုံကို ဖော်ပြပေးလိုက်ပါတယ်ဗျာ။ လေ့လာကြည့်လိုက်ကြပါအုံး

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



```

using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;

namespace IDCheck_Tutorial
{
    public partial class frmMain : Form
    {
        DataSetFolder.DsMainTableAdapters.Tbl_MainTableAdapter tblMain = new
        DataSetFolder.DsMainTableAdapters.Tbl_MainTableAdapter();

        public frmMain()
        {
            SqlConnection conn = new SqlConnection(Properties.Settings.Default.IDCheckConString);
            conn.Open();
            InitializeComponent();
        }

        private void linkSetting_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
        {
            this.Hide();
            frmSetting showSetting = new frmSetting();
            showSetting.ShowDialog();
        }

        private void picView_Click(object sender, EventArgs e)
        {
            if (PicOpenFileDialog.ShowDialog() == DialogResult.OK)
            {
                picView.ImageLocation = Path.GetFullPath(PicOpenFileDialog.FileName);
            }
        }

        byte[] ConvertImageToByte(string filesource)
        {
            FileInfo info = new FileInfo(filesource);
            FileStream Fs = new FileStream(filesource, FileMode.Open, FileAccess.Read);
            BinaryReader Br = new BinaryReader(Fs);
            byte[] returnByte = Br.ReadBytes((int)Fs.Length);
            return returnByte;
        }

        private void btnInsert_Click(object sender, EventArgs e)
        {
            try
            {
                if (txtNo.Text != "" & txtID.Text != "" & txtIssueDate.Text != "" & txtFatherName.Text != "" & txtName.Text != "" &
                txtBirthDay.Text != "" & txtReligion.Text != "" & txtHeight.Text != "" & cbBloodType.Text != "" & txtWork.Text != "" &
                txtMark.Text != "" & txtAddress.Text != "" & picView.ImageLocation != "")
                {
                    tblMain.Insert_Main(txtNo.Text, txtID.Text, txtIssueDate.Text, txtFatherName.Text, txtName.Text,
                    txtBirthDay.Text, txtReligion.Text, txtHeight.Text, cbBloodType.Text.ToString(), txtWork.Text, txtMark.Text, txtAddress.Text,
                    ConvertImageToByte(picView.ImageLocation.ToString()));
                    MessageBox.Show("သိမ်းဆည်းပြီးပါပြီ", "Insert");
                }
            }
            else
            {
                MessageBox.Show("ကျေးဇူးပြုပြီး Data များကို စုံလင်ပြည့်စုံစွာဖြည့်သွင်းပါ။", "Insert");
            }
        }
    }
}

```

Setting ကိုနှိပ်လိုက်တဲ့အခါ frmMain Form ကြီးပျောက်သွားပြီး frmSetting Form ပေါ်လာအောင် ရေးထားတာပါ။

Picture Box ကိုနှိပ်ပြီး မိမိစက်ထဲက Image တွေကို ရွေးယူနိုင်အောင်ပါ။

Image Type ကို Binary Type အဖြစ်ပြောင်းပြီး သိမ်းနိုင်အောင်ပါ။

Tbl\_Maing Table ကိုထဲကို Insert လုပ်ခြင်း အပိုင်းပါ။

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့...



```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void frmMain_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
}

```

Run ထားတဲ့ Program ကြီးမှ အပြီးတိုင် ထွက်သွားနိုင်အောင်ပါ။

ဒီ Insert လုပ်ခြင်းအပိုင်းမှာ အားနည်းချက်လေးတွေ ရှိနေပါသေးတယ်။ ဘာလည်းဆိုရင်တော့ ကျနော်တို့ Insert လုပ်တဲ့နေရာမှာ ဘာ Filter မှ စစ်မထားပါဘူး။ ဒါကြောင့် ကျနော်တို့ Database အတွင်းကို တူညီတဲ့ Record ဘာနှစ်ကြောင်းရှိက်ထည့်ထည့် Program က Error မပြဘဲ လက်ခံနေအုံးမှာပါ။ ဒီလိုမဖြစ်အောင် ဘာလို ဖြေရှင်းနိုင်လည်း ဆိုတာကိုတော့ ရှေ့လေ့ကျင့်ခန်းတွေမှာ ကျနော်ဆွေးနွေးခဲ့ပြီးလဲ ဖြစ်ပါတယ်။ မိတ်ဆွေတို့ အနေနဲ့ ကိုယ်တိုင် ဆက်လေ့ကျင့်ကြည့်ကြပေါ့ဗျာ။ ဒါမှမဟုတ်ဘူးဆိုရင်လည်း ကျနော် နောက်အပတ် လေ့ကျင့်ခန်းလေးမှာ ဆက်လက်ဆွေးနွေးပေးသွားပါ့မယ်။

စာတွေလဲ အရမ်းရှည်နေပြီဆိုတော့ ဒီအပတ် ကျနော်တို့ရဲ့ လေ့လာခြင်းကို ဒီမှာပဲ ခဏရပ်ထားလိုက် ကြရအောင်ဗျာ။ နောက်အပတ်လေ့ကျင့်ခန်းလေးမှာတော့ Insert အတွက် Filter စစ်ခြင်း၊ Tbl\_Main Table ထဲမှ Data တွေကို DataGridView ထဲမှာ ပေါ်အောင်လုပ်ခြင်း၊ ထည့်သွင်းပြီး Record များကို Delete လုပ်ခြင်း၊ ... စတဲ့ အပိုင်းလေးတွေကို ဆက်လက်ပြီး ဆွေးနွေးပေးသွားပါ့မယ်လို့ ပြောကြားရင်း..... မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမအားလုံး လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေဗျာ။

C# and SQL Server 2008 အားလေ့လာခြင်း သင်ခန်းစာ အခန်း(4)လေးပဲဖြစ်ပါတယ်။ ဒီသင်ခန်းစာများကို ကျွမ်းကျင်လို့ ရေးသားခြင်းတော့မဟုတ်ပါဘူး။ မိမိလေ့လာမိသမျှလေးတွေကို လေ့လာသူများ၊ စာဖတ်သူများအနေဖြင့် မြန်မာလိုလေး ဖတ်ရှုရခြင်းကြောင့် တစ်လုံး၊ တစ်ပါးလောက် စိတ်ထဲရောက်သွားစေချင်တဲ့ ရည်ရွယ်ချက်နဲ့ ရေးသားဖော်ပြပေးခြင်းဖြစ်ပါတယ်။ ကျနော်ကိုယ်တိုင်လည်း လေ့လာခဲ့ဖြစ်လို့ တင်ပြပုံ မှားယွင်းခဲ့တာ၊ အဓိပ္ပာယ်ကွဲလွဲခဲ့တာများ ရှိကောင်းရှိနိုင်ပါလိမ့်မယ်။ ဒါကြောင့် သည်းခံဖတ်ရှုပေးကြဖို့နှင့် ကျနော်၏အားနည်းချက်များကို ပြန်လည်ထောက်ပြပေးကြပါလို့.... လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။

**Tay Zar Lin**  
**Koyinmaung007@gmail.com**  
**Programmingknowledge.blogspot.com**

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။