

Guía para entender SQL Server Y consejos

Índice

Crear base de datos	2
Procedimientos almacenados para cada caso	3
Vistas	4
Estructura	5
Conexión.....	5
FORM	6
Form objetivo	7
CRUD	7
DATAGRID	11
Colores	13
JOB.....	14

Crear base de datos

```
-- Crear la base de datos
CREATE DATABASE Visual_Basic_CRUD;
GO

-- Usar la base de datos
USE Visual_Basic_CRUD;
GO

-- Crear la tabla Estado
CREATE TABLE Estado (
    EstadoID INT PRIMARY KEY IDENTITY(1,1),
    Estado NVARCHAR(50) NOT NULL
);
GO

-- Crear la tabla Tienda
CREATE TABLE Tienda (
    TiendaID INT PRIMARY KEY IDENTITY(1,1),
    Nombre NVARCHAR(100) NOT NULL,
    Ubicacion NVARCHAR(100) NOT NULL,
    EstadoID INT NOT NULL,
    FOREIGN KEY (EstadoID) REFERENCES Estado(EstadoID)
);
GO

-- Crear la tabla Empleado
CREATE TABLE Empleado (
    EmpleadoID INT PRIMARY KEY IDENTITY(1,1),
    Nombre NVARCHAR(100) NOT NULL,
    TiendaID INT NOT NULL,
    EstadoID INT NOT NULL,
    FOREIGN KEY (TiendaID) REFERENCES Tienda(TiendaID),
    FOREIGN KEY (EstadoID) REFERENCES Estado(EstadoID)
);
GO

-- Crear la tabla Producto
CREATE TABLE Producto (
    ProductoID INT PRIMARY KEY IDENTITY(1,1),
    Nombre NVARCHAR(100) NOT NULL,
    EstadoID INT NOT NULL,
    FOREIGN KEY (EstadoID) REFERENCES Estado(EstadoID)
);
GO

--Inserts para estado
INSERT INTO Estado(Estado) VALUES (1);
INSERT INTO Estado(Estado) VALUES (2);
```

Procedimientos almacenados para cada caso

```
CREATE PROCEDURE SP_AGREGAR_EMPLEADO
@NOMBRE NVARCHAR(100),
@TIENDA INT
AS
BEGIN
INSERT INTO Empleado (Nombre,TiendaID,EstadoID)
VALUES (@NOMBRE, @TIENDA,1);
END;
GO

CREATE PROCEDURE SP_BUSCAR_EMPLEADO
@Buscar NVARCHAR(100)
AS
BEGIN
SELECT
Empleado.EmpleadoID AS [#],
Empleado.Nombre AS [Nombre],
Empleado.TiendaID as [ID_de_la_tienda],
Tienda.Nombre AS [Nombre_de_la_Tienda]
FROM
Empleado INNER JOIN
Tienda ON Empleado.TiendaID = Tienda.TiendaID WHERE Empleado.EstadoID = 1 AND (CAST
(Empleado.EmpleadoID AS nvarchar) + Empleado.Nombre + Tienda.Nombre) Like '%' +
@Buscar + '%';
END;
GO

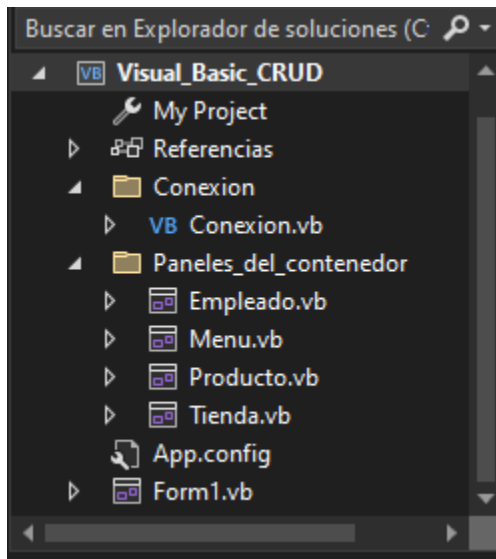
CREATE PROCEDURE SP_MODIFICAR_EMPLEADO
@EMPLEADOID INT,
@NOMBRE NVARCHAR(100),
@TIENDAID INT
AS
BEGIN
UPDATE Empleado SET
EMPLEADO.Nombre = @NOMBRE,
EMPLEADO.TiendaID = @TIENDAID
WHERE EMPLEADO.EmpleadoID = @EMPLEADOID
END;
GO

CREATE PROCEDURE SP_ELIMINAR_EMPLEADO
@EMPLEADOID INT
AS
BEGIN
UPDATE Empleado SET
EMPLEADO.EstadoID = 2
WHERE EMPLEADO.EmpleadoID = @EMPLEADOID
END;
GO
```

Vistas

```
CREATE VIEW V_MOSTRAR_EMPLEADO AS
SELECT
Empleado.EmpleadoID AS [#],
Empleado.Nombre AS [Nombre],
Empleado.TiendaID as [ID de la tienda],
Tienda.Nombre AS [Nombre Tienda]
FROM
Empleado INNER JOIN
Tienda ON Empleado.TiendaID = Tienda.TiendaID WHERE Empleado.EstadoID = 1;
GO
```

Estructura



Conexión

```
'LIBRERIAS NECESARIAS
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Module Conexion

    'CADENA DE CONEXION'
    Public conexionsql As New SqlConnection("Server=DESKTOP-V683HQU\SQLEXPRESS;
    Database=Visual_Basic_CRUD; User Id=sqlserver; Password=root;")

    'ABRIR Y CERRAR CONEXIONES'

    Sub Abrir_Conexion()
        If conexionsql.State = 0 Then
            conexionsql.Open()
        End If
    End Sub

    Sub Cerrar_Conexion()
        If conexionsql.State = 1 Then
            conexionsql.Close()
        End If
    End Sub

End Module
```

FORM

```
Public Sub CargarFormularioEnPanel(formulario As Form)
    ' Limpia el panel antes de agregar un nuevo formulario
    Contenedor.Controls.Clear()

    ' Establece el tamaño del formulario para que se ajuste al panel
    formulario.TopLevel = False
    formulario.FormBorderStyle = FormBorderStyle.None
    formulario.Dock = DockStyle.Fill

    ' Agrega el formulario al panel
    Contenedor.Controls.Add(formulario)
    formulario.Show()
End Sub
```

```
Private Sub Button1_MouseClick(sender As Object, e As MouseEventArgs) Handles
Button1.MouseClick

    Dim frmEmpleado As New Empleado()

    ' Llamar a la función para cargar el formulario en el panel
    CargarFormularioEnPanel(frmEmpleado)

End Sub
```

Form objetivo

1. Libreria

```
Imports System.Data.SqlClient
```

2. Limpiar txt

```
Private Sub Limpiar()  
    txtNombreEmpleado.Clear()  
    txtIDUbucion.Clear()  
End Sub
```

3. Funcionalidad de botón agregar y cerrar

```
Private Sub Button1_MouseClick(sender As Object, e As MouseEventArgs) Handles  
Button1.MouseClick  
  
    PanelAgregar.Visible = True  
    Limpiar()  
    btnAgregar.Enabled = True  
    btnModificar.Enabled = False  
    txtNombreEmpleado.Focus()  
  
End Sub
```

```
Private Sub Button4_MouseClick(sender As Object, e As MouseEventArgs) Handles  
Button4.MouseClick  
  
    Limpiar()  
    PanelAgregar.Visible = False  
    btnAgregar.Enabled = True  
    btnModificar.Enabled = True  
  
End Sub
```

CRUD

1. AGREGAR

```
Private Sub btnAgregar_MouseClick(sender As Object, e As MouseEventArgs)  
Handles btnAgregar.MouseClick  
  
    Dim Consulta As New SqlCommand  
  
    If txtNombreEmpleado.Text <> "" And txtIDUbucion.Text <> "" Then  
        Try  
  
            Abrir_Conexion()  
            Consulta = New SqlCommand("SP_AGREGAR_EMPLEADO", conexionsql)  
            Consulta.CommandType = 4  
  
            Consulta.Parameters.AddWithValue("@Nombre",  
txtNombreEmpleado.Text.ToString)
```

```

        Consulta.Parameters.AddWithValue("@TIENDA",
Convert.ToInt32(txtIDUbucacion.Text))

        Consulta.ExecuteNonQuery()

        Cerrar_Conexion()

        PanelAgregar.Visible = False
        Limpiar()

        Mostrar()

    Catch ex As Exception

    End Try

Else
    MsgBox("Los capos son obligatorios")

End If

End Sub

```

2. PROGRAMACION PARA MOSTRAR

```

Sub Mostrar()
    Dim dt As New DataTable
    Dim da As SqlDataAdapter

    Try
        Abrir_Conexion()
        'Forma para traer la data desde una vista'
        da = New SqlDataAdapter("SELECT * FROM V_MOSTRAR_EMPLEADO", conexionsql)

        'Forma para traer la data desde un procedimiento'
        'da = New SqlDataAdapter("SP_Mostrar_Tienda", conexionsql)
        da.Fill(dt)
        DataGridEmpleado.DataSource = dt
        Cerrar_Conexion()

        'Cambiar ancho del datagrid'
        DataGridEmpleado.Columns(0).Width = 70
        DataGridEmpleado.Columns(1).Width = 20
        DataGridEmpleado.Columns(2).Width = 200
        DataGridEmpleado.Columns(3).Width = 200
        DataGridEmpleado.Columns(4).Width = 200

        'Apariencia de los encabezados del datagrid'
        DataGridEmpleado.EnableHeadersVisualStyles = False
        Dim estilo As DataGridViewCellStyle = New DataGridViewCellStyle()
        estilo.BackColor = Color.White
        estilo.ForeColor = Color.Black
        estilo.Font = New Font("Arial", 10, FontStyle.Regular Or FontStyle.Bold)
        DataGridEmpleado.ColumnHeadersDefaultCellStyle = estilo

    Catch ex As Exception

    End Try

```



```
End Sub
```

3. MOSTRAR EN FORM AL CARGAR LA PESTAÑA

```
Private Sub Empleado_Load(sender As Object, e As EventArgs) Handles Me.Load  
    Mostrar()  
End Sub
```

4. PROGRAMACION PARA BUSCAR EN TIEMPO REAL

```
Private Sub BuscarDG()  
    Dim dt As New DataTable  
    Dim da As SqlDataAdapter  
  
    Try  
  
        Abrir_Conexion()  
        'Forma para traer la data desde una vista'  
        da = New SqlDataAdapter("SP_BUSCAR_EMPLEADO", conexionsql)  
        da.SelectCommand.CommandType = 4  
  
        da.SelectCommand.Parameters.AddWithValue("@Buscar", txtBuscar.Text)  
  
        da.Fill(dt)  
        DataGridEmpleado.DataSource = dt  
        Cerrar_Conexion()  
  
        'Cambiar ancho del datagrid'  
        DataGridEmpleado.Columns(0).Width = 70  
        DataGridEmpleado.Columns(1).Width = 20  
        DataGridEmpleado.Columns(2).Width = 200  
        DataGridEmpleado.Columns(3).Width = 200  
        DataGridEmpleado.Columns(4).Width = 200  
  
        'Apariencia de los encabezados del datagrid'  
        DataGridEmpleado.EnableHeadersVisualStyles = False  
        Dim estilo As DataGridViewCellStyle = New DataGridViewCellStyle()  
        estilo.BackColor = Color.White  
        estilo.ForeColor = Color.Black  
        estilo.Font = New Font("Arial", 10, FontStyle.Regular Or  
FontStyle.Bold)  
        DataGridEmpleado.ColumnHeadersDefaultCellStyle = estilo  
  
        Catch ex As Exception  
  
        End Try  
  
End Sub
```

5. ACTIVAR FUNCIONALIDAD DE BUSCAR EN TIEMPO REAL

```
Private Sub txtBuscar_TextChanged(sender As Object, e As EventArgs) Handles  
txtBuscar.TextChanged  
    BuscarDG()
```

```
End Sub
```

6. AL DARLE DOBLE CLICK QUE META LOS DATOS EN LOS TXT

```
Private Sub DataGridEmpleado_DoubleClick(sender As Object, e As EventArgs)
Handles DataGridEmpleado.DoubleClick
    PanelAgregar.Visible = True
    btnAgregar.Enabled = False
    btnModificar.Enabled = True

    Try
        Dim ID As String

        TextID.Text = DataGridEmpleado.SelectedCells.Item(1).Value
        txtNombreEmpleado.Text = DataGridEmpleado.SelectedCells.Item(2).Value
        txtIDUbucion.Text = DataGridEmpleado.SelectedCells.Item(3).Value

    Catch ex As Exception

    End Try

End Sub
```

7. MODIFICAR

```
Private Sub btnModificar_MouseClick(sender As Object, e As MouseEventArgs)
Handles btnModificar.MouseClick

    Dim Consulta As New SqlCommand

    If txtNombreEmpleado.Text <> "" And txtIDUbucion.Text <> "" Then
        Try
            Abrir_Conexion()
            Consulta = New SqlCommand("SP_MODIFICAR_EMPLEADO", conexionsql)
            Consulta.CommandType = 4

            Consulta.Parameters.AddWithValue("@EMPLEADOID",
TextID.Text.ToString)
            Consulta.Parameters.AddWithValue("@NOMBRE",
txtNombreEmpleado.Text.ToString)
            Consulta.Parameters.AddWithValue("@TIENDAID",
txtIDUbucion.Text.ToString)

            Consulta.ExecuteNonQuery()

            Cerrar_Conexion()

            PanelAgregar.Visible = False
            Limpiar()

            Mostrar()

        Catch ex As Exception

        End Try
    End If
End Sub
```

```

Else
    MsgBox("Los capos son obligatorios")

End If

End Sub

```

8. ELIMINAR

```

Private Sub DataGridViewEmpleado_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridViewEmpleado.CellContentClick

    'Verificar si se ha dado click sobre la columna eliminar'
    If e.ColumnIndex = DataGridViewEmpleado.Columns.Item("Eliminar").Index Then
        Dim result As DialogResult
        result = MsgBox("El registro sera eliminado", vbQuestion +
vbOKCancel, "Empleados")

        If result = DialogResult.OK Then
            Dim Consulta As SqlCommand

            Try
                Abrir_Conexion()
                Consulta = New SqlCommand("SP_ELIMINAR_EMPLEADO",
conexionsql)

                Consulta.CommandType = 4
                Consulta.Parameters.AddWithValue("@EMPLEADOID",
DataGridViewEmpleado.SelectedCells.Item(1).Value)
                Consulta.ExecuteNonQuery()
                Cerrar_Conexion()

                Mostrar()

            Catch ex As Exception

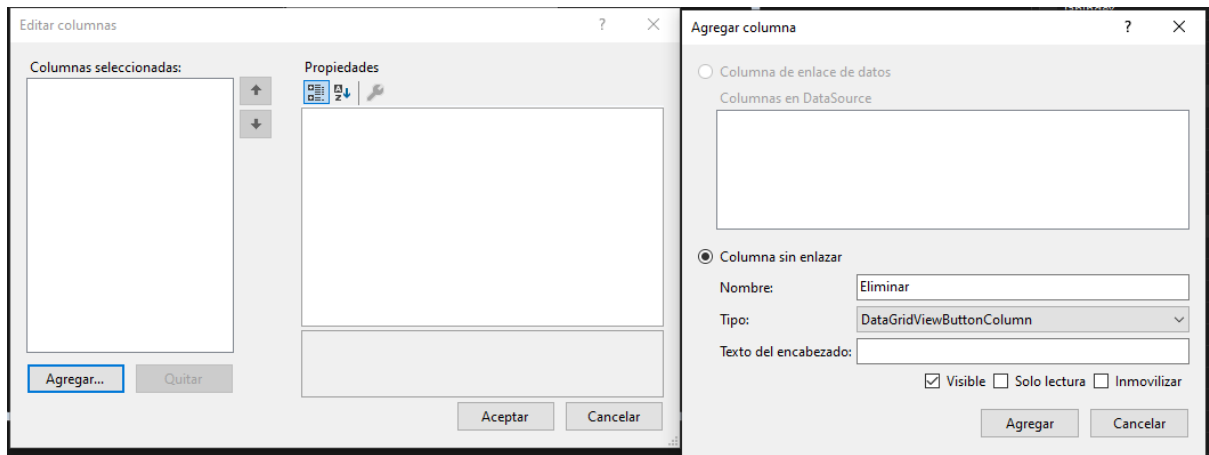
            End Try
        Else
            MsgBox("Eliminacion Cancelada")
        End If
    End If

End Sub

```

DATAGRID

Comportamiento	
AllowDrop	False
AllowUserToAddRows	False
AllowUserToDeleteRows	False
AllowUserToOrderColumns	False
AllowUserToResizeColumns	False
AllowUserToResizeRows	True
ClipboardCopyMode	EnableWithAutoHeaderText
ColumnHeadersHeightSize	AutoSize
ContextMenuStrip	(ninguno)
EditMode	EditOnKeystrokeOrF2
Enabled	True
ImeMode	NoControl
MultiSelect	True
ReadOnly	True
RowHeadersWidthSizeMode	EnableResizing
SelectionMode	FullRowSelect
StandardTab	False
TabIndex	0
TabStop	True
VirtualMode	False
Visible	True
Datos	
Diseño	
(Name)	DataTienda
Anchor	Top, Bottom, Left, Right
AutoSizeColumnsMode	None
AutoSizeRowsMode	None
Dock	None
GenerateMember	True
Location	16; 19
Locked	False
Margin	3; 3; 3; 3
MaximumSize	0; 0
MinimumSize	0; 0
Modifiers	Friend
RowHeadersWidth	41
ScrollBars	Both
Size	777; 464
Foco	
Varios	
Columns	(Colección)



Colores

Gris claro

31; 31; 31



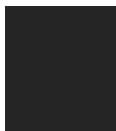
Gris oscuro

26; 26; 26

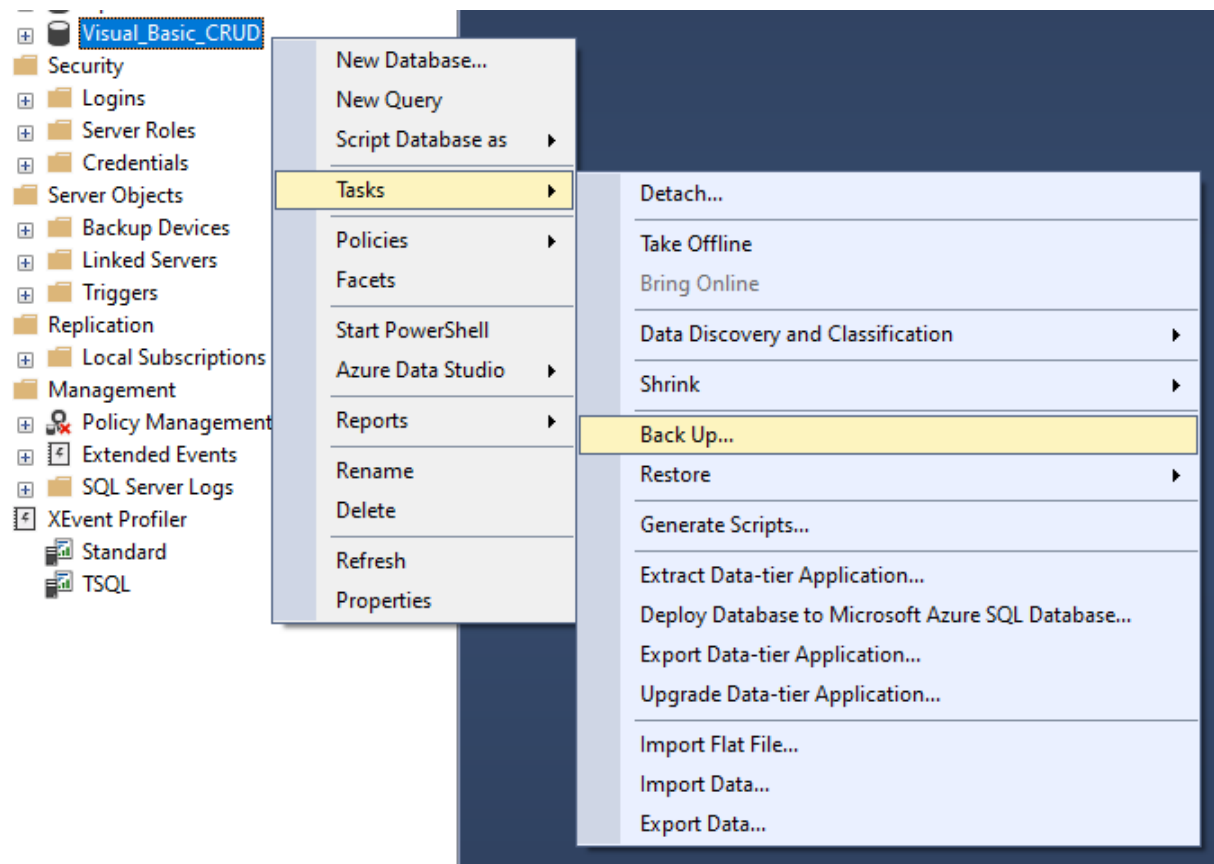


Mouse encima del botón

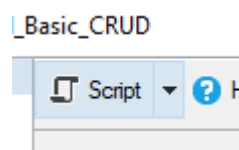
37; 37; 37



1. Obtener el SCRIPT

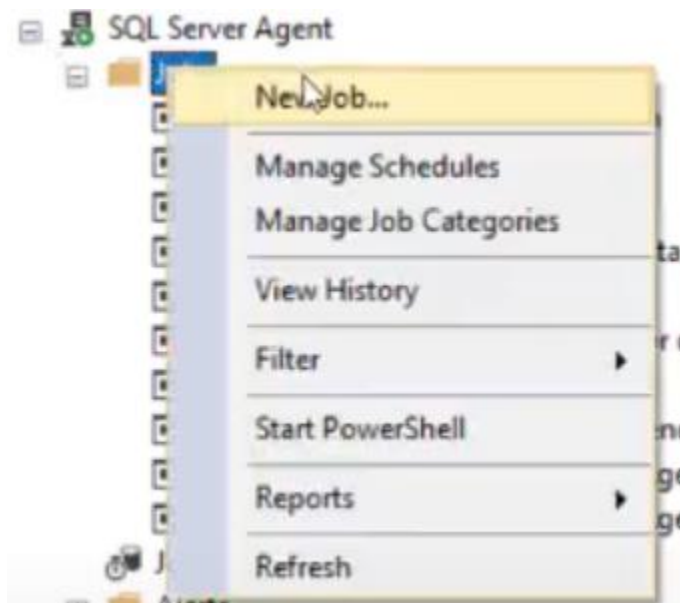


2. Darle a script para que me genere el script

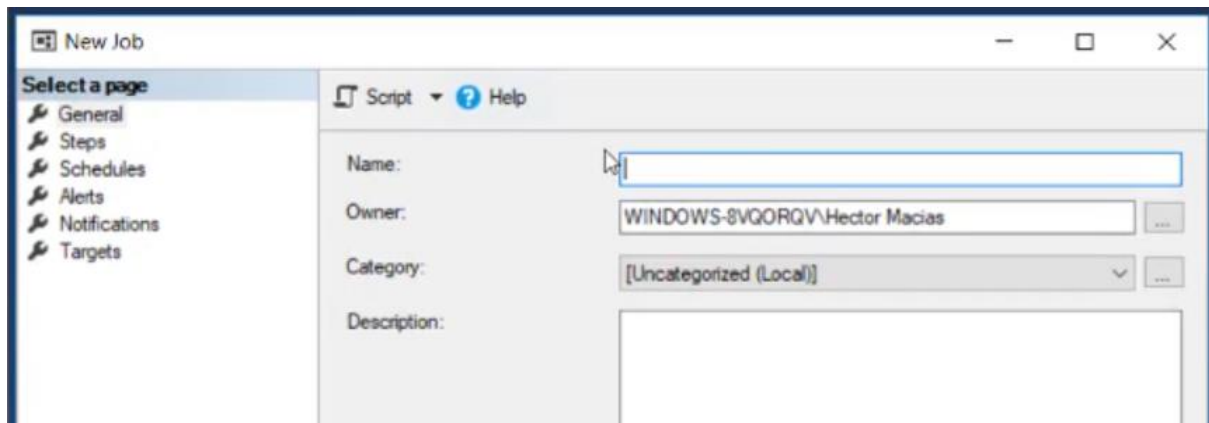


3. Compiar el script

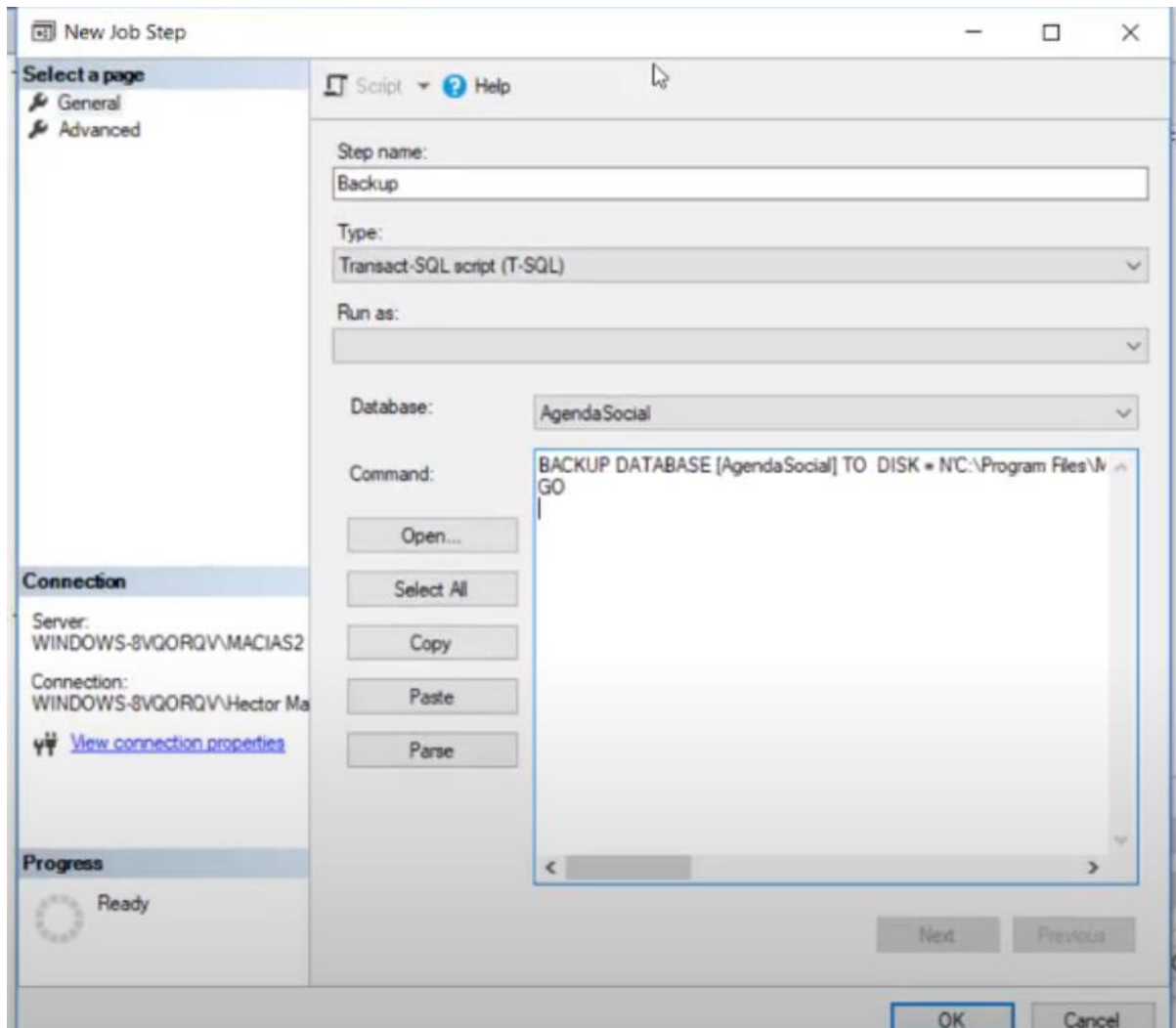
4. Crear el job



5. Ponerle nombre al JOB



6. STEPS -> NEW



- Step name
- Database
- Código que genere anteriormente
 - Parse para probarlo

7. Schedule

Job Schedule Properties - Backup

Name: Backup Jobs in Schedule

Schedule type: Recurring ☐ Enabled

One-time occurrence

Date: 17/09/2018 Time: 11:59:45

Frequency

Occurs: Weekly

Recurs every: 1 week(s) on

☒ Monday ☐ Wednesday ☐ Friday ☐ Saturday
☐ Tuesday ☐ Thursday ☐ Sunday

Daily frequency

☒ Occurs once at: 8:00:37
☐ Occurs every: 1 hour(s) Starting at: 12:00:37 Ending at: 12:00:37

Duration

Start date: 17/09/2018 ☐ End date: 17/09/2018 ☒ No end date:

Summary

Description: Occurs every week on Monday at 8:00:37. Schedule will be used starting on 17/09/2018.

OK Cancel Help

Programar recurrencia, días, hora,