

Delphi, conexión a firebird con componentes IBX

Introducción

Este tutorial pretende mostrar la facilidad de conectarse a una base de datos Firebird usando Turbo Delphi, sin embargo, debería de funcionar con otras versiones usando las pestañas IBX o Interbase.

Esta primera parte solo se va a enfocar a hacer una conexión a través de una simple consulta SQL y mostrar los datos de una tabla.

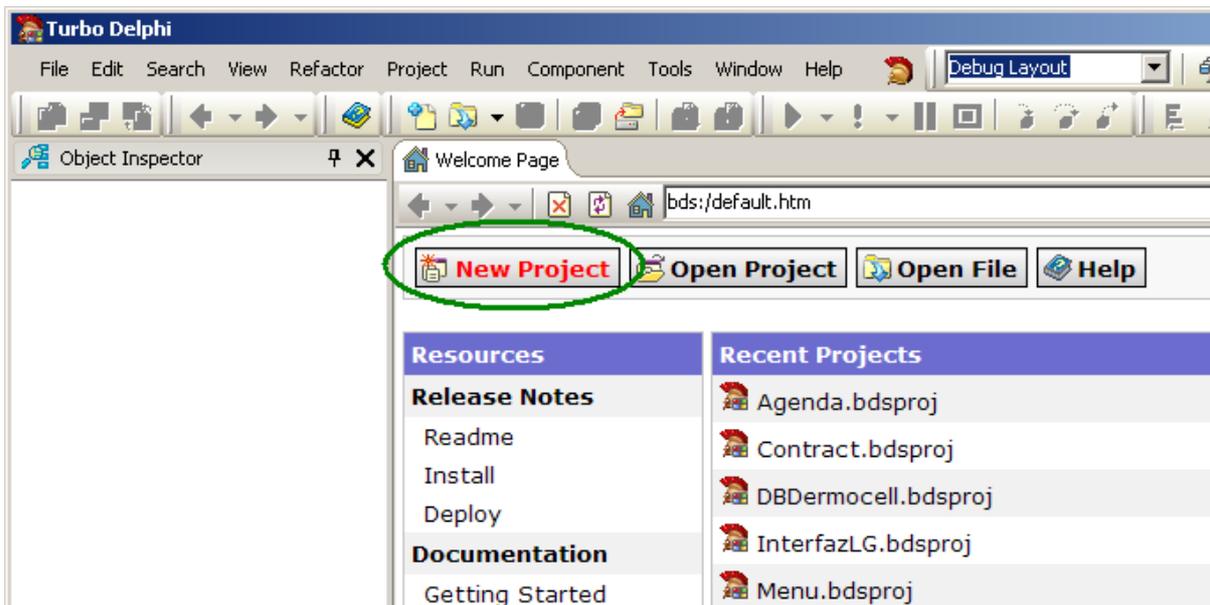
En los tutoriales siguientes haremos cosas mas interesantes y productivas (al menos eso espero).....

He dejado los nombres de los componentes en su forma original para mayor comprensión de este tutorial, sin embargo, es recomendable que asignes nombres acordes a tu proyecto para evitar confusiones.

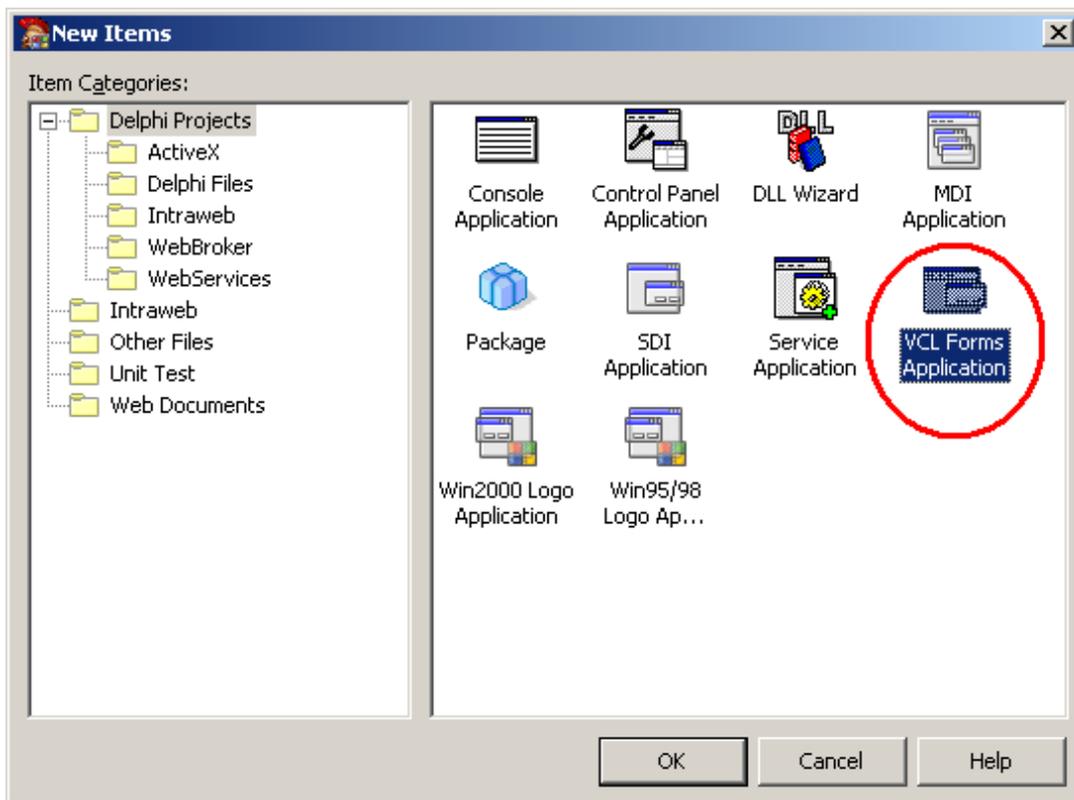
Nota: Asumo que ya tienen una base de datos Firebird creada para realizar este ejercicio.

Preparando la forma visual

Comencemos creando un nuevo proyecto en el IDE de Turbo Delphi presionando el enlace "New Project" o desde el Menú File-->New



Veremos una ventana donde vamos a seleccionar una aplicación VCL dentro de la categoría Delphi Projects.



A continuación "pegaremos" los siguientes componentes en la forma que ya tenemos abierta y decida donde los colocará.

De la Paleta Interbase



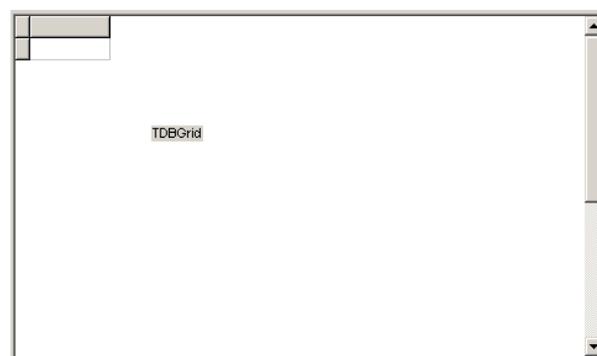
De la Paleta Data Access



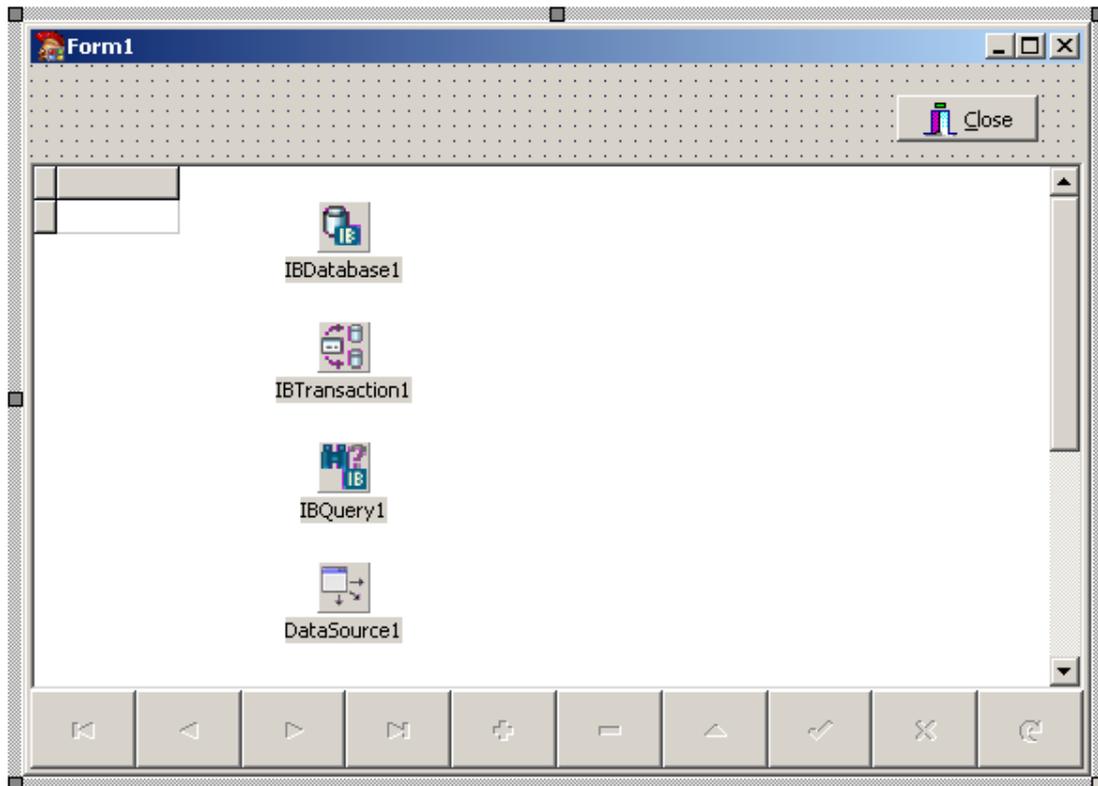
De la Paleta Additional



De la Paleta Data Controls



Veamos un ejemplo de como se vería una vez que se ha decidido donde estará cada componente.



Con esto hemos terminado la parte visual, como ven hasta ahora no hemos escrito nada de código en nuestro proyecto.

Preparando la conexión con la base de datos

En este capítulo vamos a ver como se realiza la conexión con la base de datos usando los componentes que agregamos a la forma.

Abrimos la pantalla de Editor del componente IBDatabase1, dando doble click del mouse.

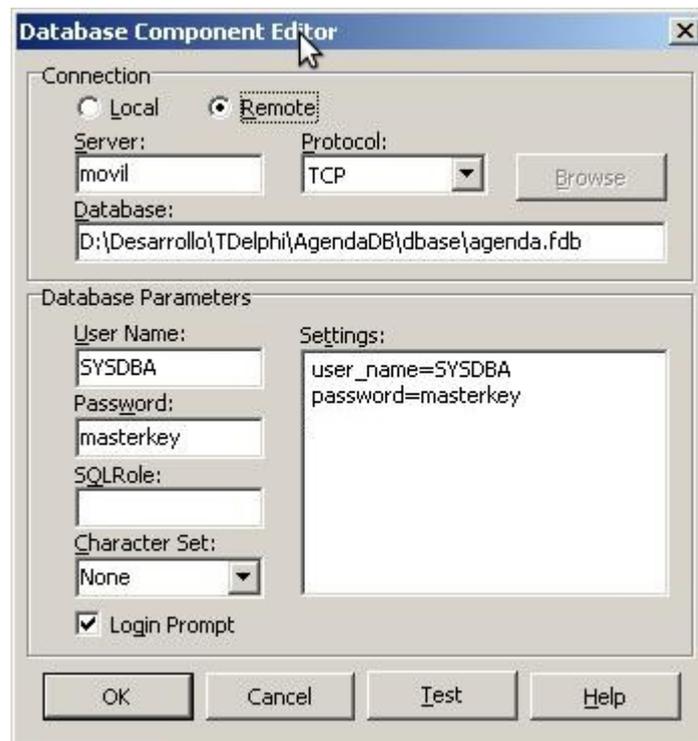
En esta ventana realizaremos lo siguiente:

1. Conexión

Elegimos conexión remota (me parece la mejor opción)
Asignamos el Servidor (en este caso mi maquina es MOVIL)
Asignamos el Protocolo (TCP)
Asignamos la base de datos (Ruta\base.fdb)

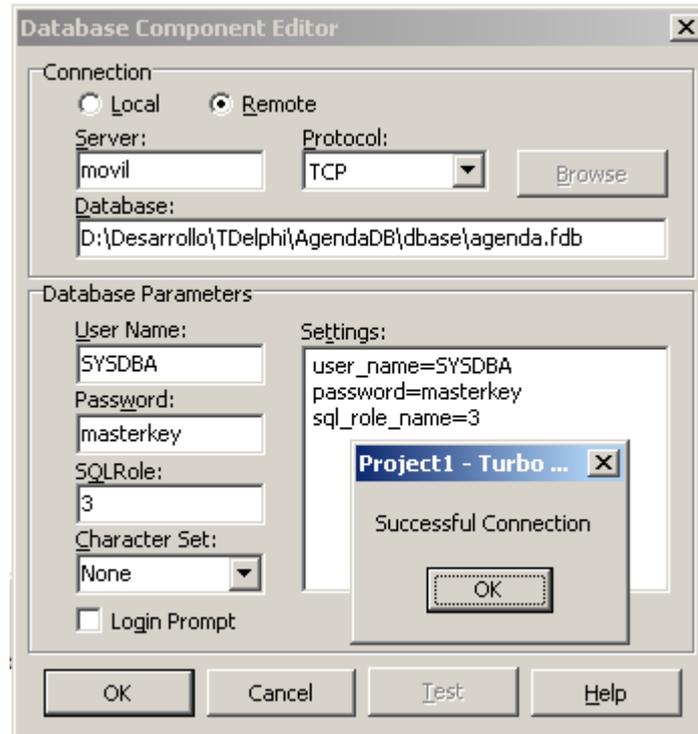
2. Parámetros

Usuario (SYSDBA)
Contraseña (masterkey)
Role (dejar en blanco)
Conjunto de carácter (None)



3. Probar conexión

Probemos la conexión presionando el botón Test y si es correcta nos mostrará la siguiente ventana.



Si todo esta bien la conexión será exitosa, en caso contrario habrá que verificar los parámetros asignados.

Una vez que te asegures que esta conexión es exitosa ya estamos listos para acceder a los datos.

Generar la sentencia SQL y ejecutar el programa

Pues bien, ya le dimos forma a nuestro proyecto, ahora corresponde realizar una simple consulta SQL para acceder a los datos y mostrarlos en el DBGrid.

Para ello, necesitamos asignar las propiedades de cada componente para "ligarlos", esto lo haremos de la siguiente manera:

Asignamos en la propiedad DefaultTransaction del componente IBDataBase1 el componente IBTransaction1

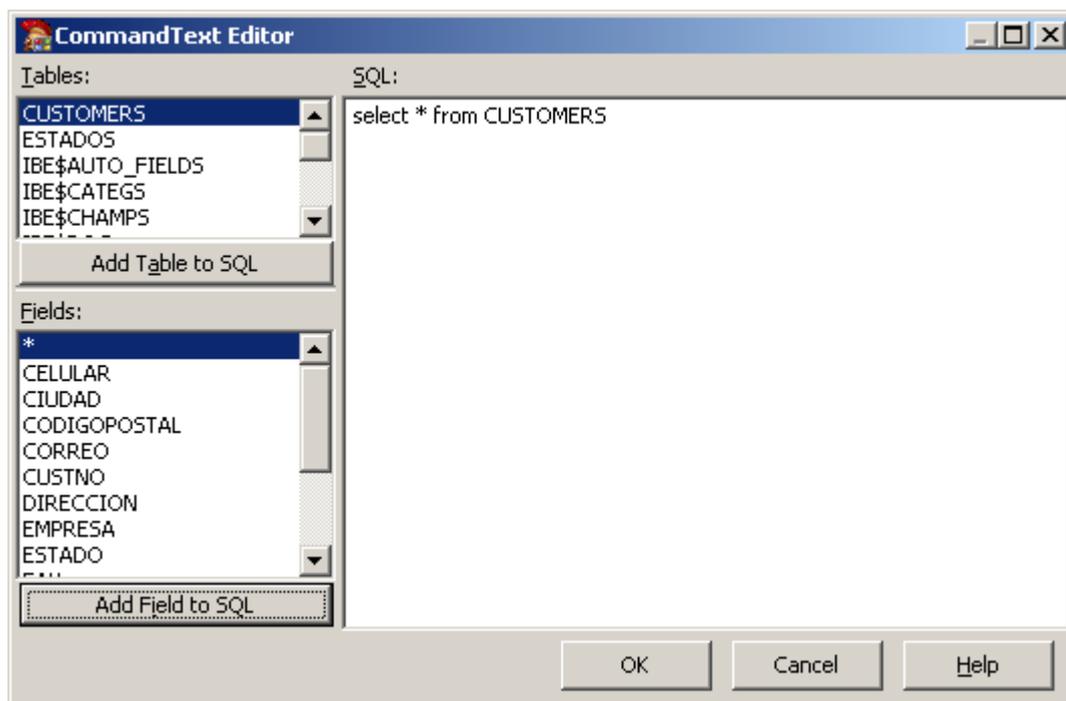
Asignamos en la propiedad DataBase del componente IBQuery1 el componente IBDataBase1

Asignamos en la propiedad DataSet del componente DataSource1 el componente IBQuery1

Asignamos en la propiedad DataSource del componente DBGrid1 el componente DataSource1

Asignamos en la propiedad DataSource del componente DBNavigator1 el componente DataSource1

En este momento ya estamos listos para generar una consulta en el componente IBQuery1, para ello demos click derecho del mouse y seleccionamos EditSQL o bien en la propiedad SQL del componente nos mostrará la siguiente ventana :



En esta ventana podemos escribir directamente en el editor SQL la sentencia que deseamos o de lo contrario seleccionamos la tabla que deseamos, presionamos el botón Add Table to SQL, nos mostrará la sentencia básica del SELECT, posteriormente seleccionamos (*) o los campos que deseamos y presionamos al botón Add Fields to SQL.

Presionamos el botón OK y ya está lista nuestra consulta.

Asigne la propiedad Kind del TBitBtn que agregamos a la Forma como bkClose.

Ahora, solo nos falta ejecutar la consulta, usted puede decidir en que momento la ejecutará, puede ser por ejemplo en el evento OnShow de la Forma.

Código: [\[Seleccionar\]](#)

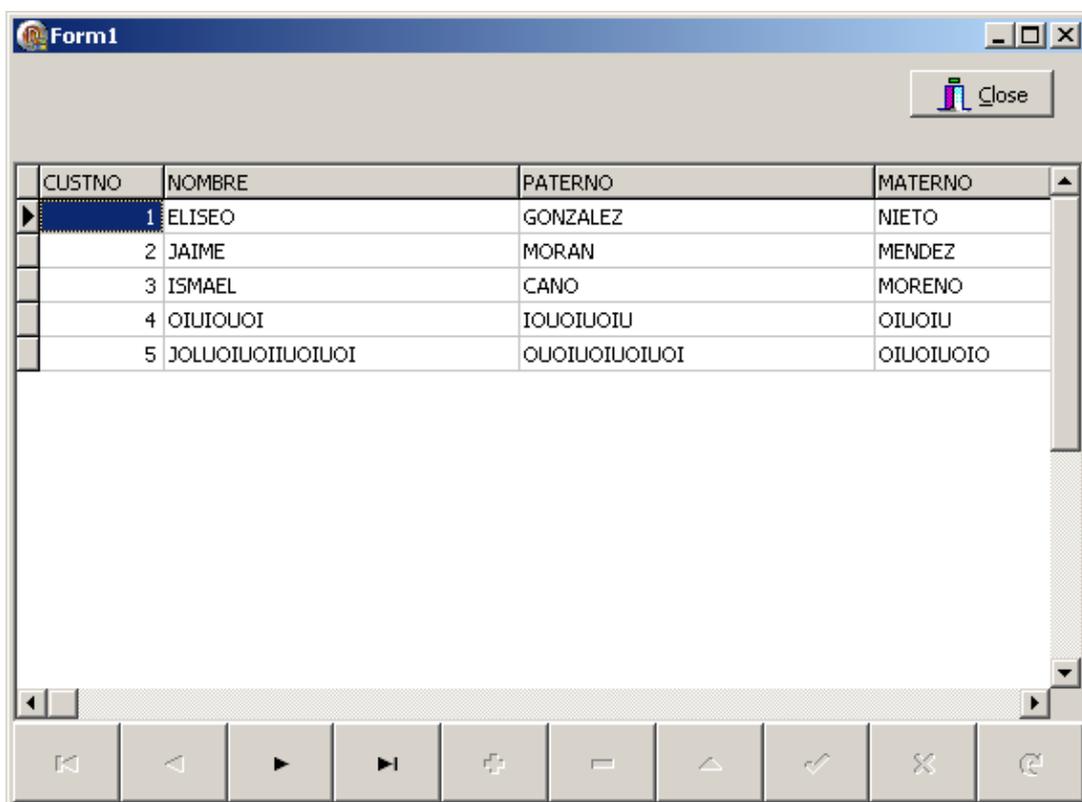
```
procedure TForm1.FormShow(Sender: TObject);  
begin  
    IBQuery1.Open;  
end;
```

Para cerrar la consulta use el evento OnClose de la Forma.

Código: [\[Seleccionar\]](#)

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    IBQuery1.Close;  
end;
```

Compile y ejecute el programa y verá algo como esto:



Ahora podrás recorrer el DataSet usando el DBNavigator el cual sólo tendrá habilitados los botones de movimiento dentro del DataSet, los botones de edición no ya que estamos usando una simple consulta, en el siguiente tutorial veremos como agregar, modificar y borrar los registros de la base de datos.

Con esto hemos terminado el primer tutorial de conexión con Firebird desde Turbo Delphi, como ves fue algo muy sencillo de hacer y con solo dos líneas de código.

Continuamos desarrollando los tutoriales y en esta segunda parte, como ya les había mencionado, vamos a entrar en conceptos mas específicos.

En este tutorial aprenderemos como agregar, modificar y borrar registros a través de nuestra aplicación que ya hemos creado en el primer tutorial de esta serie y que solo nos mostraba los datos de la tabla en la forma.

Preparar la forma

Para desarrollar este ejemplo vamos a usar el componente IBDataSet que se encuentra en la pestaña Interbase (IBX en otras versiones de Delphi) colocándolo en nuestra forma y eliminamos el componente IBQuery que usamos anteriormente.



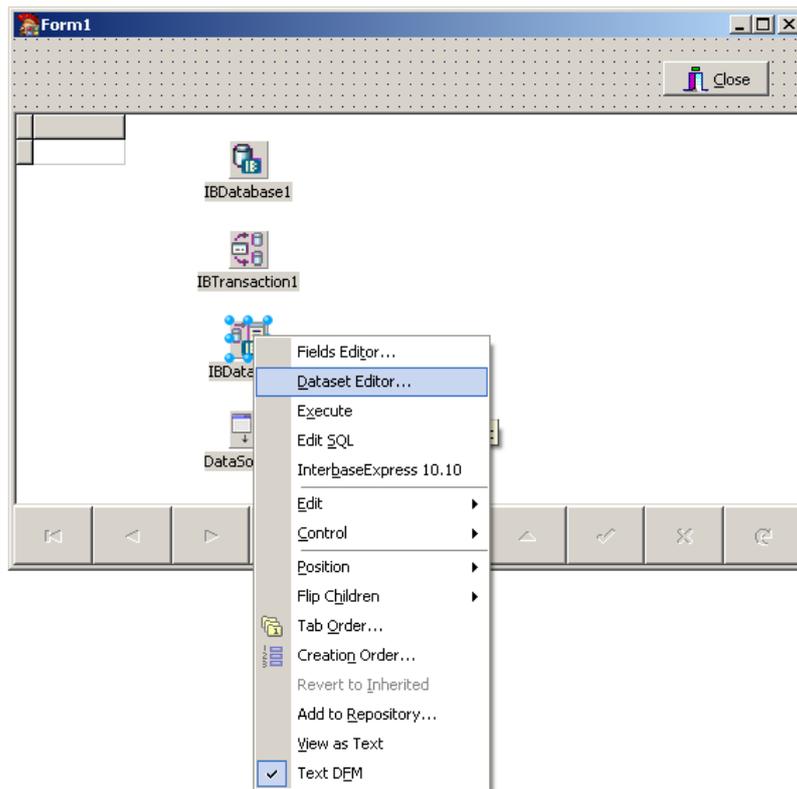
Modificar las ligas de los componentes

Una vez que ya hemos eliminado el componente IBQuery y agregamos el componente IBDataSet, tenemos que ligarlos entre ellos, siga estos pasos:

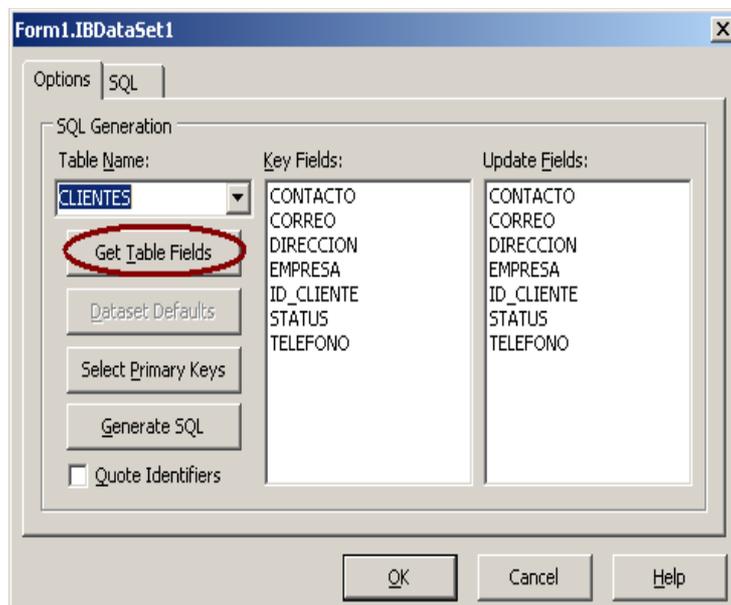
*Asignamos en la propiedad DataBase del componente IBDataSet el componente IBDataBase1
Asignamos en la propiedad DataSet del componente DataSource1 el componente IBDataSet*

Configurar las propiedades del IBDataSet

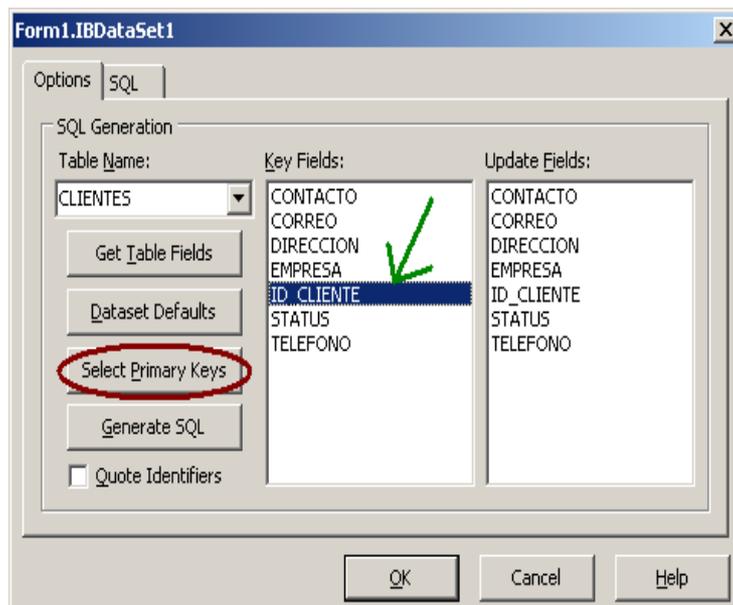
Para configurar nuestro IBDataSet, presiona el botón derecho de tu mouse sobre el componente y selecciona DataSet Editor,



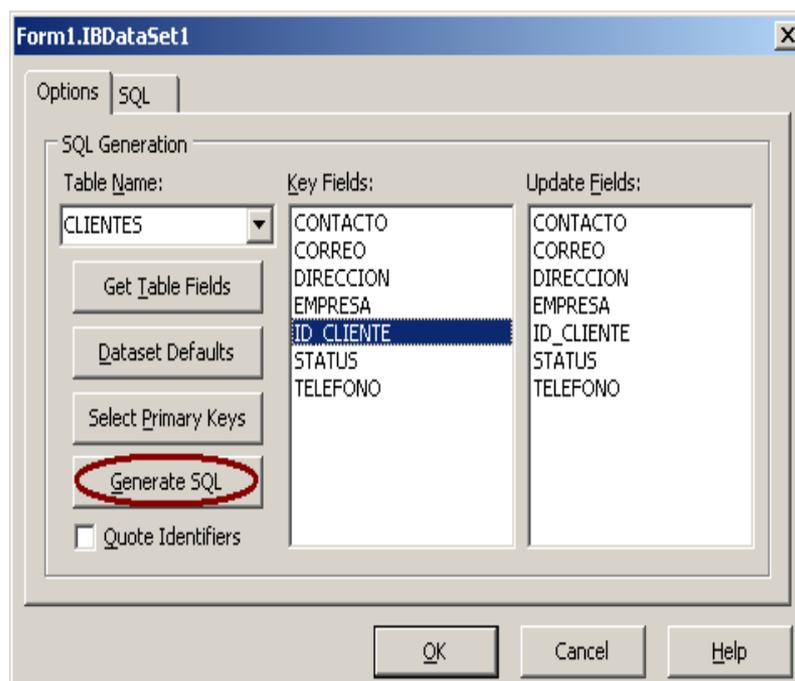
Verás la siguiente ventana, donde debes presionar el botón Get Table Fields



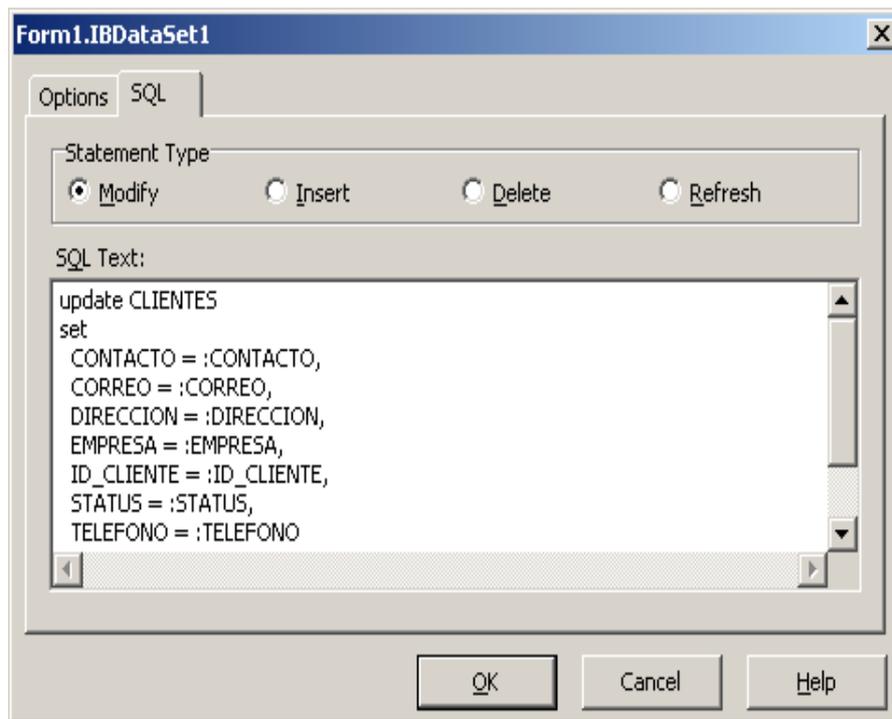
Después selecciona la llave que tengas asignada en tu tabla y presiona el botón Select Primary Keys, en este caso solo seleccionaré ID_CLIENTE que es mi llave primaria,



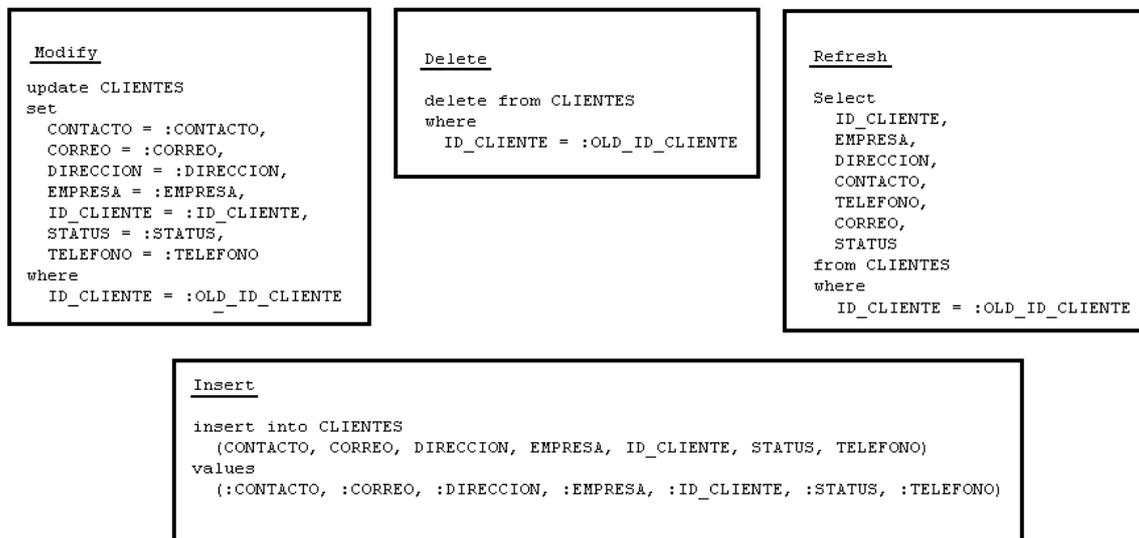
Como último paso presionamos el botón de Generate SQL para que Delphi se encargue de crear las sentencias SQL apropiadas, que bien ¿no?, todo lo hace nuestro querido Delphi.



Después de haber presionado el botón anterior nos aparecerán las sentencias SQL para cada uno de los procesos que se requieren, Modificar, Insertar, Borrar y Refrescar.



Estas son las sentencias SQL que Delphi ha creado por mi.



Ejecutar el Proyecto

Ahora, solo nos falta modificar el código que ya tenemos en nuestro proyecto en los eventos OnShow y OnClose. Solo hay que cambiar IBQuery por IBDataSet en ambos eventos.

Código: [\[Seleccionar\]](#)

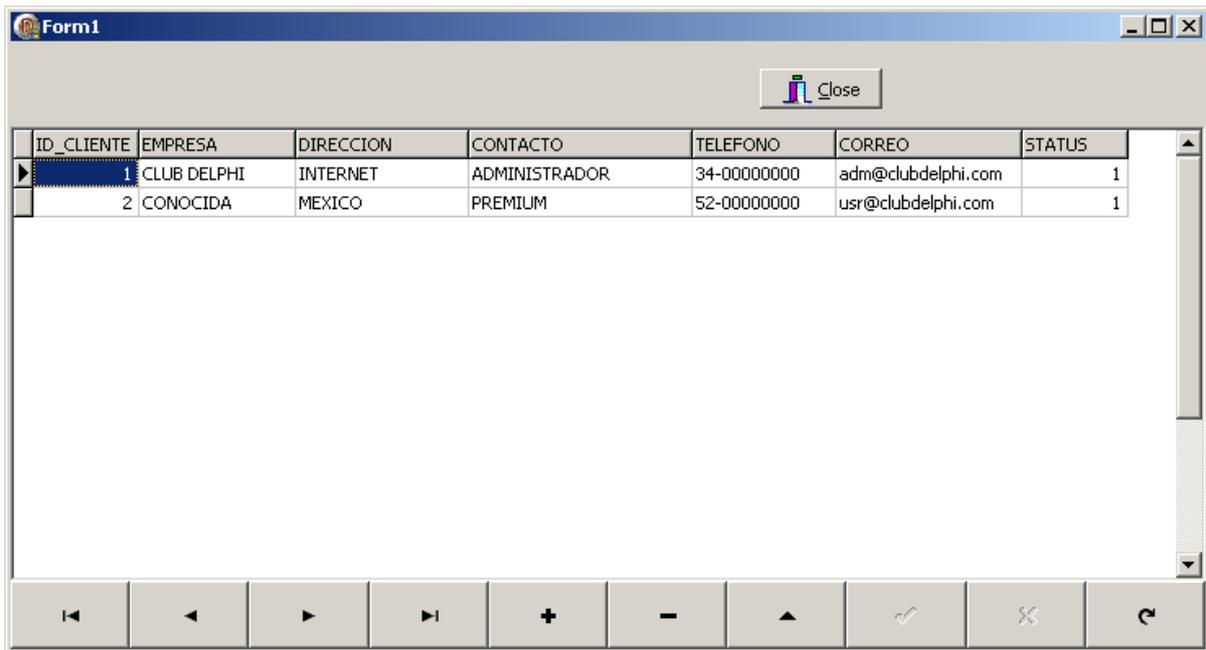
```

procedure TForm1.FormShow(Sender: TObject);
begin
  IBDataSet1.Open;
end;
  
```

Código: [\[Seleccionar\]](#)

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
    IBDataSet1.Close;  
end;
```

Ejecutamos nuestro proyecto y ya podremos realizar las operaciones deseadas.



Bueno, seguiremos trabajando con estos tutoriales, aún estamos en lo básico y no hemos contemplado algunas cuestiones propias de la base de datos, así que tengan paciencia y pronto seguiremos con esta serie de artículos.