

# **BASES DE DATOS EN VISUAL BASIC**

## **ACCESS + SQL**

*by* **Damián Sottosanti**  
*Ultima actualización:* 03-08-2004

---

### **CREAMOS LA BASE DE DATOS EN ACCESS**

*Abrimos Access y elegimos "Crear una nueva base de datos usando" "Base de datos de Access en blanco"*



*Lo guardamos, por ejemplo, en "mis documentos", con el nombre "db1.mdb"*

*Después creamos una tabla en vista diseño*



*Ingresen los mismos datos q ven a continuación, ya q estos son los datos (la tabla) q vamos a usar para programar.*

**Tabla1 : Tabla**

	Nombre del campo	Tipo de datos	Descripción
?	nombre	Texto	
	apellido	Texto	
	edad	Número	

Propiedades del campo

General | Búsqueda

Tamaño del campo: 20

Formato:

Máscara de entrada:

Título:

Valor predeterminado:

Regla de validación:

Texto de validación:

Requerido: No

Permitir longitud cero: No

Indexado: Sí (Sin duplicados)

Compresión Unicode: Sí

Un nombre de campo puede tener hasta 64 caracteres de longitud, incluyendo espacios. Presione F1 para obtener ayuda acerca de los nombres de campo.

Guarden la tabla con el nombre "Tabla1"

Ahora ingresamos los datos en la tabla:

**Tabla1 : Tabla**

	nombre	apellido	edad
	Venito	Fandulo	18
	Monica	Geronimo	69
	Pedro	Picapiedras	22
	Pamela	Astro	22
*			0

Registro: 4 de 4

Y ahora lo mas importante. Para no tener inconvenientes de compatibilidad vamos a hacer lo siguiente:

"Herramientas -> Utilidades de las base de datos -> Convertir base de datos -> A una versión anterior de la base de datos de Access..."



Guardamos con el nombre "base1.mdb"

Esta base de datos llamada "base1.mdb" es la q vamos a usar en Visual Basic. No usaremos la otra (db1) por motivos de compatibilidad. En definitiva, ambas bases de datos son iguales, solo cambia la versión.

### AHORA TRABAJAMOS EN VISUAL BASIC

Cuando creamos el proyecto lo primero q debemos hacer para trabajar con nuestra base de datos es la "referencia al motor de bases de datos de Microsoft". Para esto desde Visual Basic vamos al menu Proyecto -> Referencia y seleccionamos Microsoft DAO 3.51 Object Library (si tenes otra versión igual tiene q funcionar).

Ahora para ver lo q estamos haciendo cuando ejecutamos, vamos a agregar una ListBox (llamada List1) en nuestro formulario.

A partir de ahora es todo código.

Creamos las variables en la parte General del formulario:

*Dim BDD as DataBase 'Objeto para manejar la base de datos*

*Dim TBL as RecordSet 'Objeto para manejar la Tabla*

Es importante q tengan en cuenta lo q maneja cada objeto.

Dijimos q el objeto BDD maneja la base de datos. Entonces podemos cargarla allí de la siguiente manera:

*Set BDD = OpenDatabase("c:\mis documentos\base1.mdb") 'Abre la base de datos*

Bueno, ahora q tenemos abierta la base de datos, vamos a realizar unas consultas utilizando ordenes SQL. Por lo tanto podriamos declarar una variable para almacenar nuestra sentencia SQL.

*Dim SQL As String*

Ya estamos listos para aplicar SQL...

## ORDENES PARA MODIFICAR DATOS

UPDATE  
INSERT INTO  
DELETE

### UPDATE

Podemos cambiar los datos q queramos en la tabla q queramos mediante la orden UPDATE. Por ejemplo, en la tabla1, donde teniamos nombre, apellido y edad. Supongamos q queremos q todas las edades se pongan a 0 (cero).

```
Private Sub Form_Load()  
Dim BDD As Database  
Dim TBL As Recordset  
Dim SQL As String  
  
Set BDD = OpenDatabase("c:\mis documentos\base1.mdb")  
SQL = "UPDATE tabla1 SET edad = 0"  
BDD.Execute SQL  
SQL = "SELECT * FROM tabla1"  
Set TBL = BDD.OpenRecordset(SQL)  
TBL.MoveFirst  
Do Until TBL.EOF  
    List1.AddItem TBL("nombre") & " " & TBL("apellido") & " tiene " & TBL("edad")  
    TBL.MoveNext  
Loop  
  
TBL.Close  
BDD.Close  
  
End Sub
```

En este caso cambiamos la manera de trabajar, o mejor dicho, para la orden UPDATE trabajamos directamente sobre el objeto Database q almacena la base de datos donde queremos realizar el cambio. Para ejecutar una sentencia SQL podemos poner *nombre\_variable\_database.Execute "sentencia\_SQL"*

Veamos nuestra sentencia: actualizar (UPDATE) de la tabla1 las edades, ponerlas a cero (SET edad=0).

Tambien podemos utilizar la orden WHERE para especificar algo mas preciso. Por ejemplo poner a cero las edades q sean mayores q 21.

```
SQL = "UPDATE tabla1 SET edad = 0 WHERE edad>21"
```

## INSERT INTO

También podemos insertar nuevos registros. Para ello utilizamos la orden INSERT INTO. Veamos un ejemplo. En la tabla1 tengo los campos nombre, apellido y edad. Bien, ahora quiero agregar un nuevo nombre, apellido y edad.

```
SQL = "INSERT INTO tabla1 (nombre,apellido,edad) VALUES('damian','sotto',22)"  
BDD.Execute SQL
```

Bien, primero decimos insertar en la tabla1 (INSERT INTO tabla1) nuevos valores para los campos (nombre, apellido,edad) los valores son para el primer campo damian, para el segundo sotto y para el tercero 22.

En VALUES se escriben los valores de los campos en el mismo orden en q se especificaron. Los caracteres van entre ' ' y las fechas entre {}. Si no especificamos valores entonces el campo queda vacío. Pero si o si debe haber un valor para poder crear un nuevo registro.

Bueno, ya sabemos como actualizar datos y como agregar datos. Ahora nos falta como eliminar datos.

## DELETE

Esta sentencia se utiliza para borrar los registros de una tabla. La sentencia es DELETE FROM nombre\_tabla WHERE condiciones.

```
SQL = "DELETE FROM tabla1 WHERE edad<21"  
BDD.Execute SQL
```

Con esta sentencia borro todos los registros cuya edad sea menor a 21. Si no especifico un WHERE, se borran todos los registros, o sea, la tabla me queda vacía.

```
SQL = "DELETE FROM tabla1"  
BDD.Execute SQL
```

Ahora, si la tabla esta vacía, cuando hagamos una búsqueda dentro de ella el programa va a dar error. Una solución sería:

```
SQL = "SELECT * FROM tabla1"  
Set TBL = BDD.OpenRecordset(SQL)
```

```
If TBL.EOF Then "EOF esta en verdaero si no hay datos"  
    MsgBox "No hay datos que coincidan con la búsqueda especificada"  
    Exit Sub  
End If  
'si llega hasta aca es porq hay datos  
TBL.MoveFirst
```

```
Do Until TBL.EOF
List1.AddItem TBL("nombre") & " " & TBL("apellido") & " tiene " & TBL("edad")
TBL.MoveNext
Loop

TBL.Close
BDD.Close
```