



EBook Gratis

APRENDIZAJE

ms-access

Free unaffiliated eBook created from
Stack Overflow contributors.

#ms-access

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con ms-access.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
¿Qué es MS-Access y para qué lo usamos?.....	3
Motores de base de datos de Microsoft Access.....	4
Versiones.....	4
Capítulo 2: Acceder a SQL.....	5
Examples.....	5
Introducción a Access SQL.....	5
Union (Merge) Consultas.....	5
La función COUNT ().....	6
Capítulo 3: Cómo solucionar problemas de accesos.....	8
Introducción.....	8
Observaciones.....	8
Examples.....	9
Descompilar la base de datos.....	9
Prueba de memoria de computadora.....	9
Eliminar datos binarios del formulario.....	10
Eliminar campos "Objeto OLE".....	11
Reconstruir la base de datos completa.....	11
Capítulo 4: Consultas parametrizadas.....	14
Introducción.....	14
Examples.....	14
Enfoque vulnerable: cadena SQL concatenada con referencias de formulario.....	14
QueryDef Parameterized Enfoque de consulta.....	14
Capítulo 5: Tablas de auto-referencia.....	16
Observaciones.....	16
Examples.....	16

Tabla de empleados con referencia propia.....	16
Creditos.....	17

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ms-access](#)

It is an unofficial and free ms-access ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ms-access.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Comenzando con ms-access

Observaciones

Disculpas por la pared de texto. Las ediciones futuras probablemente agregarán capturas de pantalla y otros elementos visuales.

Las consultas en Microsoft Access se pueden crear por cualquiera de los cuatro métodos

1. Utilice un generador de Asistente de consultas paso a paso a través de la GUI que le hará una serie de preguntas sobre qué datos desea mostrar y cómo se relacionan los distintos bits de datos entre sí.
2. Mediante el uso de la GUI en la vista de diseño, en la que selecciona tablas y campos específicos con el mouse. El orden visual de los diversos campos se puede especificar arrastrando las columnas relevantes en el panel inferior y las propiedades adicionales para cada campo se pueden especificar en el panel Propiedades (derecha).
3. Al cambiar de la Vista de diseño a la Vista de SQL y especificar el código de consulta de SQL sin formato. En la mayoría de los casos, puede alternar libremente entre Vista de diseño y Vista de SQL: Access incorporará los cambios que realizó en una vista a la otra. La sintaxis de Access SQL es similar, pero no idéntica, a la utilizada en MySQL, PostgreSQL, Oracle o MS SQL Server (tSQL).
4. Uso de Visual Basic para la programación de aplicaciones, al que se puede acceder a través del grupo de macros de la cinta Herramientas de base de datos (Access 2007+). La manipulación de la base de datos se realiza a través de las bibliotecas ADO o DAO y utiliza la misma sintaxis que la Vista SQL en la aplicación principal, con la excepción de que se deben "escapar" varios caracteres especiales. Las consultas creadas a través de este método no son accesibles directamente desde el panel de navegación, pero deben ubicarse en una función o subprocedimiento y deben ser activadas por otros elementos (por ejemplo, mediante un botón en un formulario) a través de macros o ejecutadas directamente en la interfaz gráfica de usuario de VBA.

La edición de un valor de registro de una consulta en la vista de hoja de datos resultará en un cambio en el valor de registro subyacente, asumiendo que el campo de consulta no es una agregación o concatenación de múltiples fuentes de información.

Los formularios e informes se pueden usar para mostrar información de las consultas en forma alternativa a una simple vista de "Hoja de datos" que se parece a una hoja de cálculo de estilo Excel. Los formularios están orientados a la visualización en pantalla, mientras que los informes están dirigidos a aquellos impresos en papel.

Examples

Instalación o configuración

Microsoft Access es uno de los programas de Microsoft Office. Sin embargo, solo está disponible

en algunos paquetes de MS Office.

Si desea obtener acceso, asegúrese de examinar cuidadosamente la caja o descargar las especificaciones para cada versión de Microsoft Office. MS Access **solo** está **disponible para PC con Windows** , no está disponible en sistemas Macintosh en el entorno nativo, incluso a través de otros programas de MS Office que estén disponibles. Del mismo modo, no está disponible para sistemas operativos Linux.

En Office 365, Access se puede encontrar en las versiones Home, Personal, ProPlus, Enterprise E3 o E5, pero **no Enterprise E1 ni Business** (o B. Essentials, B. Premium).

En Office 2016 no está **incluido en los paquetes Hogar y Estudiante o Hogar y Negocio** , pero está en Profesional. No parece estar en ninguna versión para Macintosh.

¿Qué es MS-Access y para qué lo usamos?

Microsoft Access es un **generador de aplicaciones** para desarrollar bases de datos y aplicaciones basadas en datos, principalmente para uso local. Microsoft Access consta de dos elementos principales:

1. Un **sistema de administración de bases de datos relacionales (RDBMS)** que combina el [motor de base de datos Microsoft Jet](#) (Access 2003 y earlier) o el motor de base de datos de Access (Access 2007 y versiones posteriores; consulte a continuación) con herramientas de administración gráfica. Un sistema único de **tablas vinculadas** permite que las tablas remotas se traten como locales.
2. **Interfaz gráfica de usuario (GUI)** y herramientas de **desarrollo de software** , compatibles con [Visual Basic para aplicaciones \(VBA \)](#) que pueden hacer referencia a una variedad de objetos.

Es un miembro del conjunto de aplicaciones de **Microsoft Office** , incluido en las ediciones Professional y superiores o se vende por separado. Las aplicaciones de base de datos que se han creado con una versión completa de Microsoft Access pueden [compilarse para su distribución y ejecutarse a través de un Microsoft Access Runtime gratuito](#) .

Los dos elementos permiten que Microsoft Access se use de varias maneras:

- **Como base de datos** : una base de datos de Microsoft Access no requiere un **servidor de datos** , y se usa a menudo como una base de datos para aplicaciones locales, como una **base de datos de sitio web** , ubicada en el servidor web.
- **Como generador de aplicaciones de datos** : las herramientas para crear GUIs que contienen formularios y controles vinculados a tablas (*locales* o *vinculadas*) permiten a los desarrolladores crear aplicaciones locales para acceder y administrar datos locales o remotos. Los módulos VBA permiten a los desarrolladores crear capacidades no compatibles con las herramientas GUI.
- **Como un generador de aplicaciones completo** : las capacidades anteriores permiten a los desarrolladores crear aplicaciones de datos locales completos en uno o más archivos de Access.

Motores de base de datos de Microsoft Access

A través de Access 2003 (11.0), el motor de base de datos integrado era [Microsoft Jet](#) . Con Access 2007 (12.0), Microsoft introdujo un nuevo descendiente del motor Jet, el Motor de base de datos de Access (originalmente llamado Motor de conectividad de acceso y todavía conocido comúnmente como **Motor de ACE**), y lo convirtió en el predeterminado para las nuevas bases de datos. Su conjunto de características y comportamiento se superponen de forma incompleta con la última versión de Jet (4.0). Las versiones de Access lanzadas desde entonces han sido capaces de crear y trabajar con bases de datos en formato Jet (`.mdb`) o ACE (`.accdb`), a pesar de que [Jet ha sido oficialmente obsoleto](#) como tecnología.

Versiones

Microsoft Access ha existido desde 1992, y las versiones anteriores continúan su uso regular cuando se han desarrollado aplicaciones de bases de datos críticas para la empresa. Un recurso muy completo que resume el historial de lanzamiento (con enlaces a las notas de lanzamiento, donde esté disponible) es:

- [Versiones de versiones de Microsoft Access, Service Packs, revisiones y historial de actualizaciones](#)

Lea [Comenzando con ms-access en línea](#): <https://riptutorial.com/es/ms-access/topic/3609/comenzando-con-ms-access>

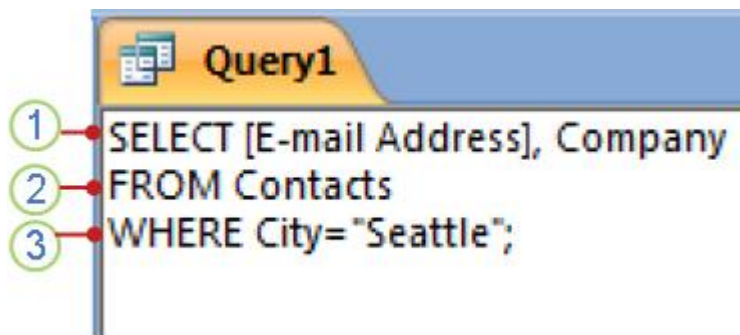
Capítulo 2: Acceder a SQL

Examples

Introducción a Access SQL

Cuando se utiliza Access, puede recuperar datos mediante consultas. Estas consultas se crean utilizando el lenguaje de consulta estructurado (SQL). Comprender SQL es importante porque puede ayudar a construir consultas mejores y más útiles.

Al crear consultas en Access, puede cambiar a "Vista SQL". Un ejemplo de una consulta de "selección" se muestra aquí:



Union (Merge) Consultas

Cuando desee combinar los resultados de varias tablas o consultas con campos similares en un solo conjunto de datos resultante sin realizar ninguna combinación relacional (es decir, si desea enumerar un conjunto de datos inmediatamente después del otro), utilizará una consulta `UNION`. Sin embargo, es notable que **estas consultas deben crearse manualmente en la vista SQL**.

La sintaxis de una consulta `UNION` es

```
SELECT
    floatingpoint_field AS floatptfld,
    text_field
FROM first_table
UNION
SELECT
    integer_field,
    decimal_field
FROM a_saved_query
UNION
SELECT
    1.0,
    "hi there Jack"
```

y devolverá un conjunto de datos de dos campos con nombres de campo (columna): `floatptfld` y `text_field`

Es fundamental que los tipos de datos (y los estilos de datos) para los campos de tablas

posteriormente combinadas sean compatibles con la primera consulta de la serie. En otras palabras, si la primera consulta `SELECT` genera un número para la primera columna, la segunda consulta también debe devolver un número en la primera columna. Además de hacer coincidir los tipos de campos en orden, las instrucciones `SELECT` deben devolver el mismo número de campos. Los nombres de los campos de la hoja de datos resultante se heredan de la primera definición de tabla.

La siguiente consulta **NO** sería legal, ya que el texto no se puede convertir en datos decimales ni los números de punto flotante se pueden convertir en números enteros (sin truncamiento o redondeo explícitos y conversión de tipos).

```
SELECT
  integer_field AS this_really_wont_turn_out_well,
  decimal_field
FROM a_saved_query
UNION
SELECT
  floatingpoint_field,
  text_field
FROM first_table
```

La función COUNT ()

Puede usar la función `COUNT ()` para devolver el número de registros que coinciden con una consulta. La siguiente tabla "Empleado" contiene los números de identificación de los empleados y el número de identificación de su gerente asociado.

ID de empleado	Manager_ID
12	37
22	37
37	63
42	45
45	63
57	45
59	45
63	

Se puede usar una declaración `COUNT ()` para averiguar cuántos empleados tienen un gerente específico:

```
SELECT COUNT(*) AS CNT FROM Employees WHERE Employee.Manager_ID = 37;
```

devoluciones

CNT

2

La función también se puede combinar en consultas más complicadas. Para saber cuántos empleados están supervisados directamente por una persona específica, se puede aplicar lo siguiente:

```
SELECT T1.Employee_ID,  
       (SELECT COUNT(*) AS CNT FROM Employees AS T2 WHERE T2.Manager_ID =  
        T1.Employee_ID) AS Supervised_Count  
FROM Employees AS T1;
```

devoluciones:

ID de empleado	Supervised_Count
12	0
22	0
37	2
42	0
45	3
57	0
59	0
63	2

La documentación de MSDN se puede encontrar [aquí](#) .

Lea **Acceder a SQL en línea**: <https://riptutorial.com/es/ms-access/topic/3985/acceder-a-sql>

Capítulo 3: Cómo solucionar problemas de accesos

Introducción

Cuando recibe un error: "Microsoft Access ha encontrado un problema y necesita cerrarse", a menudo no hay mucha información que lo ayude a identificar la causa del error. A continuación hay una serie de pasos que puede seguir para solucionar la causa de los errores.

Observaciones

Asegúrese de eliminar otras variables de la ecuación mientras prueba

Corrupción de la red

No cargue el cliente fuera de una red. Ponlo en el disco local y ejecútalo desde allí.

Construcciones corporativas

Si se encuentra en un entorno corporativo que utiliza "compilaciones de computadora" y no ha tenido éxito con la Descompilación, la Memoria de Pruebas ni la eliminación de datos Binarios, entonces rechace hacer más pruebas hasta que el equipo de TI pueda proporcionar al usuario una máquina de prueba que tenga solo Windows, Office y Service Packs instalados.

Todo el software y las actualizaciones deben instalarse a mano sin utilizar instalaciones desatendidas. No instale antivirus en esta máquina para realizar pruebas.

Comprenda que muchos departamentos de TI simplemente intentan hacer un enfoque de talla única para todas las compilaciones, y todas las compilaciones se basan en la otra. Con el tiempo, los conflictos de software pueden hacer que el acceso se bloquee o actúe de forma directa.

Mala potencia

Como se mencionó en el ejemplo de memoria, las fluctuaciones de energía pueden causar errores en la computadora. Si la base de datos está en un edificio industrial, intente obtener un acondicionador de energía o un UPS que proporcione energía limpia (sin batería, no con el varistor de óxido metálico)

Además, verifique el cable de la fuente de alimentación que está enchufado a la barra de alimentación o al tomacorriente. Asegúrese de que las especificaciones del medidor y el voltaje sean suficientes. Los departamentos de TI a menudo dejan los cables de alimentación enchufados en la estación y simplemente retiran la máquina. Después de muchos años, están utilizando fuentes de alimentación más robustas, pero no han cambiado el cable. Hacer una diferencia En caso de duda, traiga un cable nuevo, más grueso.

Examples

Descompilar la base de datos

Esta siempre debe ser su solución inicial. Una buena política es descompilar la base de datos antes de cada lanzamiento.

1. **Crea un atajo de descompilación** . Esto carga la base de datos con un modificador `"/decompile"`.
 1. Haga clic derecho en su archivo de base de datos. Seleccione Copiar
 2. Haga clic derecho en la ventana del explorador y seleccione "Pegar acceso directo"
 3. Haga clic derecho en el acceso directo y seleccione "Propiedades"
 4. En el cuadro Destino, vaya al final de la línea y agregue `/decompile`
 5. Haga clic en Aceptar para cerrar el acceso directo
2. **Abrir base de datos con Shift.**

Mantenga presionada la tecla Mayús mientras hace doble clic en este acceso directo. Esto evita que las ejecuciones automáticas se ejecuten dentro de la base de datos. Debes ir directamente a la ventana de navegación.
3. **Compactar y reparar la base de datos.** Una vez que la base de datos esté cargada, deberá hacer clic en el botón Compactar y Reparar.
 1. Localice el botón *Compactar y reparar base de datos* en la barra de **herramientas** .
 2. Mantenga presionada la tecla Shift. Manténgalo presionado mientras hace clic en el botón *Compactar y Reparar* .
4. **Recompila la base de datos**
 1. Ir a la ventana de VBA (Control + G)
 2. Seleccione Depurar -> Compilar desde el menú

Este es el proceso completo de descompilación. En general, debería corregir el 99% de todos los bloqueos de acceso o el comportamiento extraño de la forma.

Prueba de memoria de computadora

Si sus choques son aleatorios o esporádicos, siga este paso. Si sus bloqueos ocurren cada vez que ejecuta la base de datos, este paso no solucionará el problema (aunque la mala memoria puede ser la razón por la que ocurrió el daño en primer lugar).

Use un comprobador de memoria que se inicie fuera del sistema operativo y ejecute varios pases. Dos opciones populares son [MemTest86](#) (comercial) y [MemTest86 +](#) (código abierto)

Comience la prueba y déjela correr durante las horas de trabajo. La razón de esto es que otros factores en el edificio, como el ruido en los circuitos de alimentación, pueden causar errores de memoria, por lo que debe intentar mantener las variables iguales.

Si tiene errores de memoria, deberá identificar si se debe a una mala memoria en la computadora o algún otro factor. Sin embargo, esto está fuera del alcance de este documento.

Eliminar datos binarios del formulario

A veces, los bloqueos ocurren constantemente en un solo formulario o informe, o ocurren solo al imprimir. Es posible que los datos binarios dentro del formulario / informe se hayan dañado.

Guardar el objeto formulario / informe como texto Hay dos funciones no documentadas. `Application.SaveAsText` y `Application.LoadFromText`. Puede usar estas funciones para exportar las definiciones de formulario / informe, limpiar la definición y luego importarla de nuevo.

1. Haga una copia de seguridad de su base de datos antes de continuar
2. Ir a la ventana inmediata de VBA (Control + G)
3. Escriba `Application.SaveAsText acForm, "MyForm", CurrentProject.Path & "\MyForm.txt"`
(Reemplace MyForm con el nombre del formulario / informe. Use `acReport` si se trata de un informe corrupto)
4. Cambie el nombre del elemento del formulario original (por ejemplo, cambie el nombre a MyForm.Bak) dentro de la ventana de la base de datos

Limpie el archivo de definiciones de formulario / informe

1. Abra el archivo exportado (por ejemplo, MyForm.txt) en el bloc de notas
2. Elimine la línea "Checksum =" (debe estar en la línea 3)
3. Borrar datos binarios
 1. Identificar los bloques de datos binarios. Examine el archivo y verá líneas que comienzan con "Parámetro = Comenzar". Siguiendo esas líneas tendrás líneas de datos binarios codificados. Finalmente, el bloque binario terminará con una línea que consiste solo en "Fin". El bloque de datos binarios incluye la primera línea (con la instrucción Begin) y todas las líneas hasta e incluyendo la última línea (con la instrucción End).

Nota: todos estos bloques deben aparecer ANTES de las definiciones de control de formulario

2. Eliminar los bloques de datos binarios para los siguientes parámetros:

- Mapa de nombres
- PrtMip
- PrtDevMode
- PrtDevNames
- PrtDevModeW
- PrtDevNamesW

4. Busque otras cuestiones. Mientras tenga el archivo abierto, desplácese por el resto del archivo y busque cualquier cosa que le llame la atención, especialmente en el código del

módulo VBA en la parte inferior. Buscará cualquier cosa que sobresalga del resto y puede ser corrupción.

5. Guarda el archivo.

Cargue el formulario / informe de nuevo en Access y Test

1. Vuelva a cargar el formulario en Access.

- En Access, vaya a la ventana inmediata (Control + G)
- **Escriba** `Application.LoadFromText acForm, "MyForm", CurrentProject.Path & "\MyForm.txt"`
- **Decompile / Compact Repair / Recompile** (Vea el otro ejemplo en la documentación)
- **Abra el formulario / informe para probar.** Esperemos que todo esté funcionando ahora.
- **Eliminar la forma corrupta antigua** (por ejemplo, MyForm.bak)

Prevenir esta corrupción en el futuro.

La causa más común de datos binarios corruptos dentro de un informe / formulario es cuando varias computadoras / usuarios usan el mismo archivo de cliente de base de datos en lugar de tener su propia copia separada. Por esta razón, cada usuario debe tener su propio archivo de cliente en su escritorio que ejecutan.

Eliminar campos "Objeto OLE"

Si tiene imágenes u otros datos almacenados en Access como objetos OLE, debería encontrar un mejor enfoque. Cuando se almacenan los datos OLE, se almacenan de acuerdo con el software (y la versión del software) en la computadora que los almacena. Cuando otra computadora va a mostrar los datos del Objeto OLE en el formulario, pero no tiene instalada la versión / software exacto, muy a menudo esto resulta en un bloqueo de la aplicación.

Si está almacenando datos de imágenes, entonces un mejor enfoque es almacenar el nombre del archivo y, en su lugar, guardar las imágenes en una ubicación estándar. Las versiones más nuevas de acceso tienen los controles nativos para que esto sea el camino a seguir.

Reconstruir la base de datos completa

Esto es mucho trabajo, así que hazlo como último recurso después de agotar todas las demás opciones. Solo necesita hacer esto si el problema está ocurriendo para diferentes usuarios, en diferentes máquinas. Si no ocurre para todos los usuarios, lo más probable es que no sea un contenedor de base de datos corrupto.

Al igual que los pasos para eliminar datos binarios, va a reconstruir su base de datos desde cero. Este proceso es un poco ritualista, pero si se hace meticulosamente con cuidado para no "preservar" cualquier posible corrupción, el proceso es altamente efectivo.

Crear un nuevo contenedor de base de datos de acceso.

- En Access, en la pestaña Archivo, puede seleccionar "Nuevo". Crear una nueva base de datos vacía en formato ACCDB.

Mueve todos los objetos al nuevo contenedor

No utilice las funciones de importación / exportación dentro de Access para mover los objetos, y no basta con hacer clic y arrastrar. Hacer esto puede copiar los elementos corruptos al nuevo contenedor.

Mesas:

- Para cada tabla en el contenedor de acceso anterior, cree una tabla nueva en el contenedor nuevo.
- Desde la vista de diseño, copie / pegue las definiciones de campo.
- Compruebe las propiedades de la tabla para asegurarse de que coinciden en ambas bases de datos
- Mueva también las macros de datos (vea la sección de macros para saber cómo hacerlo)
- Para mover los datos, exporte los datos antiguos a XML o CSV, y luego importe desde ese formato.

Consultas:

- Cargue cada consulta en la vista SQL.
- Copiar / Pegar el texto SQL.
- Pegar en la nueva base de datos.
- Compare las propiedades de consulta para asegurarse de que coinciden.

Formularios / Informes:

- Para cada formulario / informe, use la función `Application.SaveAsText` para exportar los formularios / informes a un archivo de texto.
- Elimine los datos binarios (consulte *Eliminar datos binarios de la documentación del formulario* para familiarizarse con este proceso)
- Utilice la función `Application.LoadFromText` para volver a importar los objetos a la nueva base de datos

Macros

Tienes tres métodos para mover las macros.

1. Recrear cada macro a mano en el nuevo contenedor de la base de datos.
2. Utilice el método `Application.SaveAsText` / `Application.LoadFromText` con el parámetro `acMacro`.
3. Copiar / Pegar definiciones de macro para cada macro
 - Seleccione Todos (Control + A) para seleccionar todos los elementos de macro. Luego Copiar (Control + C).
 - Abra un documento de Bloc de notas en blanco y pegue (Control + V) la macro XML.
 - Cree una nueva macro en blanco en el nuevo contenedor de la base de datos.
 - En el Bloc de notas, seleccione Todo el texto (Control + A). Luego Copiar (Control + C)
 - En la macro en blanco, Pegar (Control + V). La macro debería aparecer. Guardalo

Módulos

- Para cada módulo, seleccione todo el código (Control + A) y pegue (Control + V) en el nuevo contenedor de la base de datos.
- Asegúrese de verificar las propiedades de la base de datos (en la ventana de VBA, vaya a Herramientas -> Propiedades del cliente)

Macros de datos

Para cada macro de datos, use los métodos SaveAsText / LoadFromText.

1. Ir a la ventana inmediata de VBA (Control + G)
2. Escriba `Application.SaveAsText acTableDataMacro, "MyTableName", CurrentProject.Path & "\MyTableName.txt"` (Reemplace MyTableName con el nombre de la tabla que contiene las macros de datos)
3. Revise el archivo en busca de signos de corrupción
4. En el nuevo contenedor de la base de datos, cargue la definición usando `Application.LoadFromText acTableDataMacro, "MyTableName", CurrentProject.Path & "\MyTableName.txt"`

Como se mencionó anteriormente, este es un **montón** de trabajo, pero tiene resultados. Este método también debe utilizarse al migrar una base de datos de Access 97 a 2000, o una base de datos de Access 2000 a 2003.

Lea **Cómo solucionar problemas de accesos en línea**: <https://riptutorial.com/es/ms-access/topic/8207/como-solucionar-problemas-de-accesos>

Capítulo 4: Consultas parametrizadas

Introducción

Las consultas parametrizadas se pueden usar para defenderse de los ataques de inyección de SQL.

Examples

Enfoque vulnerable: cadena SQL concatenada con referencias de formulario

Este es el enfoque típico para los desarrolladores novatos que crean consultas de acción SQL. Son vulnerables a los ataques de inyección SQL del tipo [Bobby Tables](#).

```
Dim strSQL As String

strSQL = "INSERT INTO Employees chrFirstName, chrLastName, chrPhone " _
        & "VALUES ('" & Me!txtFirstName & "', '" & Me!txtLastName & "', '" & Me!txtPhone &
        "');"

CurrentDb.Execute strSQL
```

QueryDef Parameterized Enfoque de consulta

Este enfoque evitará que un usuario incruste una segunda instrucción SQL en su entrada para la ejecución.

```
Dim strSQL As String
Dim db As DAO.Database
Dim qdf As DAO.QueryDef

strSQL = "PARAMETERS [FirstName] Text(255), [LastName] Text(255), [Phone] Text(255); " _
        & "INSERT INTO Employees (chrFirstName, chrLastName, chrPhone) " _
        & "VALUES ([FirstName], [LastName], [Phone]);"

Set db = CurrentDb

Set qdf = db.CreateQueryDef("", strSQL)
qdf.Parameters("FirstName") = Me!txtFirstName
qdf.Parameters("LastName") = Me!txtLastName
qdf.Parameters("Phone") = Me!txtPhone
qdf.Execute

Me!txtFirstName = vbNullString
Me!txtLastName = vbNullString
Me!txtPhone = vbNullString

qdf.Close
db.Close
Set qdf = Nothing
Set db = Nothing
```

Tipos de parámetros válidos:

- DATETIME : para las fechas (el parámetro espera la `Date` VBA)
- SHORT , LONG : para enteros (`SHORT` espera un entero, `LONG` espera un largo)
- SINGLE , DOUBLE : para punto flotante (espera `Single` y `Double` respectivamente)
- VARCHAR o TEXT : para cuerdas
- MEMO o LONGTEXT : para cadenas de más de 255 caracteres

Lea Consultas parametrizadas en línea: <https://riptutorial.com/es/ms-access/topic/9121/consultas-parametrizadas>

Capítulo 5: Tablas de auto-referencia

Observaciones

En el ejemplo anterior, se puede usar un campo de referencia (SupID) para indicar la identificación del supervisor de ese empleado.

Usar algo tan simple como un DLOOKUP puede devolver el nombre de ese supervisor.

p.ej. `DLOOKUP("Name","EmployeeTable", "ID = " & SupID)`

Otro buen ejemplo de esto es observar cómo se crean las centralitas automáticas de Access y, más específicamente, la estructura de la tabla de centralita. Cada opción de la centralita se refiere a otra opción dentro de la misma tabla, similar a la forma en que este ejemplo se refiere a sí mismo.

Examples

Tabla de empleados con referencia propia

ID	Name	SupID
1	Jim Browski	3
2	Sally Valley	3
3	Boss Hawg	
4	Yvette Deer	6
5	Mary C Lyte	6
6	Chuck Dee	
7	Rick Slick	6
8	Doug Fresh	3

Lea Tablas de auto-referencia en línea: <https://riptutorial.com/es/ms-access/topic/7992/tablas-de-auto-referencia>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con ms-access	Brad , Community , Hynek Bernard , jcb , marlan , mpag
2	Acceder a SQL	LiamH , mpag , SandPiper
3	Cómo solucionar problemas de accesos	DHW , user3728595
4	Consultas parametrizadas	MoondogsMaDawg , serakfalcon
5	Tablas de auto-referencia	geeFlo