

Unidad Didáctica 7
**Ordenación, archivos,
índice e indexación**

Contenido

1. Creación de una base de datos ordenada
2. Problemática de las bases de datos ordenadas
3. Índices e indexación de tablas
4. Relación entre tablas
5. Crear una relación
6. Integridad referencial

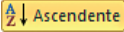
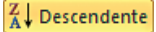
1. Creación de una Base de Datos Ordenada

Teniendo los datos ordenados nos resultará más sencillo hacer una consulta, por ejemplo, sobre la información almacenada en las tablas de una base de datos.

Si queremos ordenar la información de cualquier campo de una tabla, debemos realizar los siguientes pasos:

En primer lugar abriremos la tabla en vista **hoja de datos**.

A continuación pulsamos sobre el campo con el que queremos definir el orden. Por ejemplo, para ordenar nombres alfabéticamente, pulsamos directamente sobre la columna que contenga los nombres.

El orden puede definirlo de manera ascendente o descendente. Para ordenar los registros en orden ascendente, es decir, de la A a la Z o de menor a mayor, lo más rápido es pulsar el botón **Orden ascendente**  **Ascendente** situado en el grupo de opciones **Ordenar y Filtrar**. Si queremos ordenar los registros de forma descendente, esto es, de mayor a menor (o de la Z a la A) debemos pulsar el botón **Orden descendente**  **Descendente**. Podremos hacer lo mismo en el menú **contextual** que aparece al pulsar con el botón derecho del ratón sobre una columna, o bien con las opciones correspondientes del menú de columna que aparece al hacer clic sobre la flecha en el título del campo.

1.1. Criterios de Ordenación

Tenemos la posibilidad en *Access*, no sólo de ordenar una tabla por un único campo, sino de ordenarla igualmente por varios campos.

Pasaremos a continuación a estudiar la ordenación por dos o más campos y para ello debemos comenzar por abrir en primer lugar la tabla en vista hoja de datos. En segundo lugar, es recomendable que desplace los campos que quiera ordenar hasta la parte izquierda de la pantalla, pues *Access* los ordena de izquierda a derecha.

La tercera acción que debemos realizar, será seleccionar las columnas sobre las que quiera definir el proceso. Para ello éstas deben ser consecutivas y debemos pulsar en la primera columna y utilizar la tecla Mayúsculas para incluir el resto de las columnas.

Por último, pulsamos el botón **Orden ascendente** u **Orden descendente** de la barra de herramientas, según lo que quiera realizar.

Nombre	Apellidos
Alfredo	Maldonado Guerra
Alfredo	Fuentes Espinosa
Almudena	Benito
Antonio	Bermejo
Antonio	Gratacós Solsona
Armando	Pinto
Arturo	López
Avelino	López García
Begoña	Hurtado

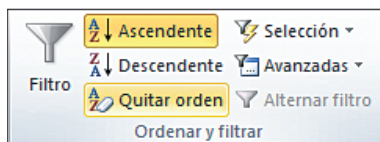
Tabla ordenada por el campo nombre.

Nombre	Apellidos
Alfredo	Fuentes Espinosa
Alfredo	Maldonado Guerra
Almudena	Benito
Antonio	Bermejo
Antonio	Gratacós Solsona
Armando	Pinto
Arturo	López
Avelino	López García
Begoña	Hurtado

Tabla ordenada por los campos nombre y apellidos.

Una vez que hemos realizado las ordenaciones que nos sean precisas no tendremos más que cerrar la pestaña de la hoja de datos con lo que Access guardará los cambios que hemos realizado.

Si por algún motivo, más tarde, decidimos quitar la ordenación realizada debemos pulsar el botón Quitar orden del grupo de opciones Ordenar y Filtrar.



2. Problemática de las Bases de Datos Ordenadas

Físicamente, los registros de una tabla se graban por orden de llegada, es decir, según se introducen. Debido a esto, ordenar físicamente una tabla suele resultar lento y poco práctico, ya que el ordenador debemos intercambiar los datos de los registros hasta haberlos ordenado completamente. La ordenación física de los registros implica que los datos se cambien de posición hasta quedar ordenados.

Cuando la base de datos con la que estamos trabajando contiene demasiada información, el proceso de ordenación de sus datos puede conllevar mucho tiempo y trabajo para la máquina.

Uno de los problemas consiste en que a medida que vamos introduciendo la información, se hace necesaria la tarea de renovar periódicamente los datos, para poder mostrarlos ordenadamente. Esto es consecuencia de la grabación secuencial de los registros, ya que si hemos ordenado los datos y posteriormente introducimos más registros, éstos se añadirán al final de la tabla, resultando que ésta ya no está ordenada, con lo que necesita una nueva ordenación.

Por este motivo, conforme tengamos una serie de nuevos registros incluidos en la tabla, debemos repetir el proceso de ordenación que desempeñamos en un principio si queremos que la tabla se muestre siempre ordenada, lo cual puede ralentizar nuestro trabajo.

Por otro lado, puede ocurrir que el proceso de ordenación sea lento, aunque esto depende en gran medida de la velocidad de nuestro ordenador, así como de la cantidad de datos que contenga la tabla.

La solución a estos problemas nos la ofrecen los índices, que pasaremos a explicar en el siguiente punto.

3. Índices e Indexación de Tablas

Como ya sabemos, la ordenación física de una base de datos implica que los datos sean cambiados de posición hasta quedar ordenados, de modo que una tabla quedará ordenada cambiando el orden de los registros físicamente. Así la tabla queda modificada como se puede ver en la siguiente figura (orden ascendente de apellidos).

Posición	Nombre	Apellidos
1	Antonio	Zambrana
2	Pedro	Muñoz
3	Juan	Muñoz
4	José	Álvarez

Tabla original.

Posición	Nombre	Apellidos
1	José	Álvarez
2	Juan	Muñoz
3	Pedro	Muñoz
4	Antonio	Zambrana

Tabla ordenada.

En el caso de la ordenación mediante índices, la ordenación no recoloca los registros, si no que crea un pequeño fichero con el orden adecuado. Si tocamos como ejemplo la misma tabla de la figura anterior, al realizar la ordenación mediante índices de la tabla, el resultado sería el siguiente:

Posición	Nombre	Apellidos
1	Antonio	Zambrana
2	Pedro	Muñoz
3	Juan	Muñoz
4	José	Álvarez

Tabla original.

Nueva Posición	Nº de Registro
1	4
2	2
3	3
4	1

Índice (Apellidos).

Según el índice, el nuevo primer registro es el 4º, el siguiente es el 2º, etc. Así, tenemos la tabla ordenada sin necesidad de cambiar los registros de lugar, ya que, cuando se pida un listado de alguna forma, se seguirá el orden marcado por el índice y no el de la tabla.

Una vez indexada la base de datos, el gestor acude al índice para consultar cuál es el siguiente registro que se va a tratar. Según el índice del ejemplo, el primer registro es el número 4, por lo que el gestor se dirigirá a la tabla y mostrará ese registro. A continuación, iremos de nuevo al índice para consultar el siguiente registro a mostrar, repitiéndose todo el proceso para cada registro de la tabla.

La ordenación mediante índices se diferencia además de la ordenación física en que el índice siempre estará actualizado, es decir, cada vez que se introducen nuevos registros, aunque físicamente un registro nuevo irá al final de la tabla, el índice se reconstruye automáticamente para tener en cuenta los nuevos registros, quedando la tabla ordenada. De este modo si en la tabla de

ejemplo introducimos un nuevo registro, la tabla y el índice quedarían de la siguiente forma:

Posición	Nombre	Apellidos
1	Antonio	Zambrana
2	Pedro	Muñoz
3	Juan	Muñoz
4	José	Álvarez
5	Jesús	Martín

Índice (Apellidos).

Nueva Posición	Nº de Registro
1	4
2	5
3	2
4	3
5	1

Tabla original.

Los índices además son automáticos y transparentes, a todos los efectos, para el usuario, es decir, el índice está ahí todo el tiempo pero no nos damos cuenta ya que no afecta en nada a nuestro trabajo ni somos conscientes de sus operaciones. Sin embargo, todas las operaciones que parecen sencillas al usuario, son el resultado de una gran cantidad de operaciones por parte del ordenador para reconstruir el índice según sea necesario. De todas formas, casi todas estas operaciones son cálculos y ordenaciones en memoria del ordenador, por lo que son mucho más rápidas que las realizadas en la ordenación física que requieren muchas operaciones de disco, cosa que deviene en un significativamente importante tiempo de operación.

Hay que decir que cuando creamos un índice sobre un campo cuyos valores pueden repetirse (como en el caso de los apellidos), nos encontramos con otro problema, y es que el gestor no ordenará estos datos idénticos, si no que los situará juntos pero en el mismo orden en que ya aparecían en la tabla. En nuestro ejemplo Pedro Muñoz y Juan Muñoz aparecen, una vez indexada la tabla, en el mismo orden en que aparecían antes de indexarla, es decir, Pedro antes que Juan.

Cuando esto ocurre, es necesario elegir un segundo campo por el que clasificar la tabla. En nuestro ejemplo podemos incluir el campo **Nombre** en el índice, con lo que el gestor cuando encuentre varios apellidos iguales, ordenará esos registros por el campo **Nombre**. El resultado sería, en nuestro ejemplo, que los registros 3 y 4 del índice se intercambiarían la posición, ya que el dato Juan, debemos ir antes que Pedro (por orden alfabético). De modo que nuestro índice quedaría así:

Posición	Nombre	Apellidos
1	Antonio	Zambrana
2	Pedro	Muñoz
3	Juan	Muñoz
4	José	Álvarez
5	Jesús	Martín

Tabla original.

Nueva Posición	Nº de Registro
1	4
2	5
3	3
4	2
5	1

Índice (Apellidos).

Los índices pueden estar así formados por varios campos, ordenándose primero la tabla por los primeros campos indicados. Pero también podemos tener varios índices, es decir, para conseguir el resultado del ejemplo anterior, habríamos creado un índice compuesto por dos campos, primero el campo **Apellidos** y luego el campo **Nombre**.

Pero si lo que necesita es, por ejemplo, poder ordenar por nombre únicamente, tendríamos que crear un índice con sólo el campo **Nombre**, teniendo en cuenta siempre que la tabla se ordenará en primera instancia por el índice referido al campo que aparezca más a la izquierda.

Posición	Nombre	Apellidos
1	Antonio	Zambrana
2	Pedro	Muñoz
3	Juan	Muñoz
4	José	Álvarez
5	Jesús	Martín

Tabla original.

Nueva Posición	Nº de Registro
1	1
2	5
3	4
4	3
5	2

Índice (Nombre).

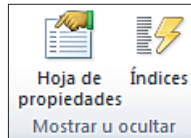
El índice **Apellidos y Nombre** está creado, pero al aparecer el campo nombre en primer lugar, la tabla se ordenará por el índice **Nombre**.

El **campo clave** es un tipo especial de índice que obligatoriamente exige unicidad, es decir, dos registros no pueden tener el mismo dato en el campo clave, de modo que el campo clave identifica inequívocamente a todos y cada uno de los registros. El campo clave también es llamado **clave principal** o **clave primaria**. Si vamos a tener relaciones entre tablas en nuestra base de datos (lo estudiaremos en el siguiente apartado) es imprescindible que nuestras tablas posean un campo clave.

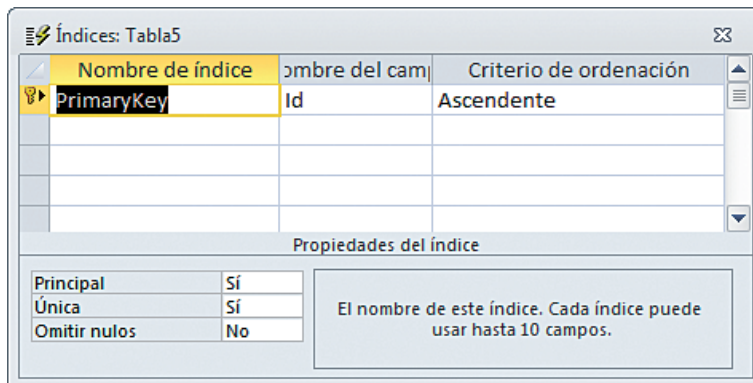
En nuestro ejemplo no nos serviría ninguno de los campos de la tabla, ya que son susceptibles de repetirse, por lo que se debemos incluir por ejemplo, el campo **DNI**, que identifica a cada persona de forma única, aunque tenga el mismo nombre y apellidos que otra.

3.1. Crear Índices en las Tablas de Access

Para crear índices en nuestras tablas de una base de datos de Access, lo definiremos normalmente en la creación de las tablas, aunque podemos también hacerlo una vez creadas, e incluso cuando ya incluyen datos.

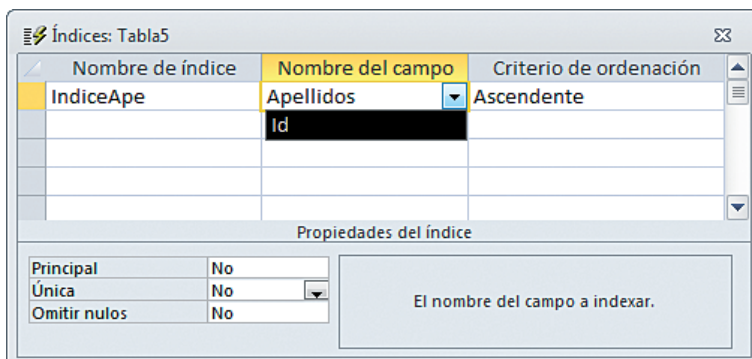


Para empezar debemos acceder a la vista de diseño. Una vez en la vista **Diseño**, sólo tenemos que pulsar sobre el botón **Índices** del grupo de opciones **Mostrar u Ocultar**, con lo que nos aparecerá la ventana de índices donde podemos ver los índices que tengamos creados y en su caso, crear, modificar y eliminar índices.



Una vez en la ventana de índices sólo tendremos que nombrar nuestro índice en el campo **Nombre de índice**, seleccionamos un campo de la tabla en la lista desplegable del campo **Nombre del campo** y finalmente elegiremos un tipo de ordenación en la lista desplegable del campo **Criterio de Ordenación**, que nos ofrece la opción de ordenar de forma ascendente o descendente. En la zona inferior izquierda de esta ventana también se nos ofrecen más opciones con

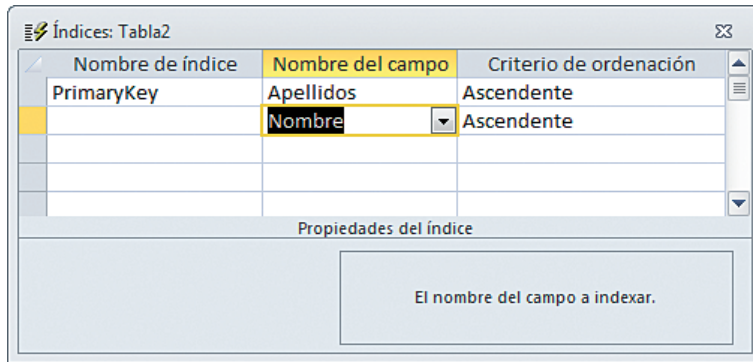
las que podremos indicar si es el campo clave (**Principal**), si exige unicidad en los datos (no se aceptan repeticiones) en el campo **Única**, o si se puede dejar el campo vacío en **Ignorar Nulos**; cuando la opción está en No, no podremos dejar el campo sin rellenar.



Cuando terminemos de crear nuestro índice, sólo tendremos que cerrar esta ventana y la de diseño guardando los cambios y nuestra tabla quedará indexada.

Nombre	Apellidos
José	Álvarez
Jesús	Martín
Pedro	Muñoz
Juan	Muñoz
Antonio	Zambrana
*	

Para indexar la tabla con un índice que ordene por más de un campo a la vez, únicamente tendremos que añadir campos al índice en las siguientes filas sin indicar el nombre del índice, con lo que conseguiremos un índice ordenado por más de un campo.

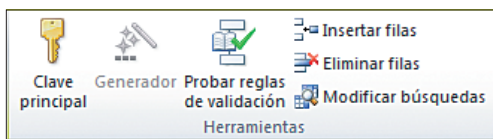


Así nuestra tabla quedará ordenada atendiendo primero al orden del primer campo, después, si se dan repeticiones en el primer campo, se ordenarán por el segundo, y así sucesivamente.

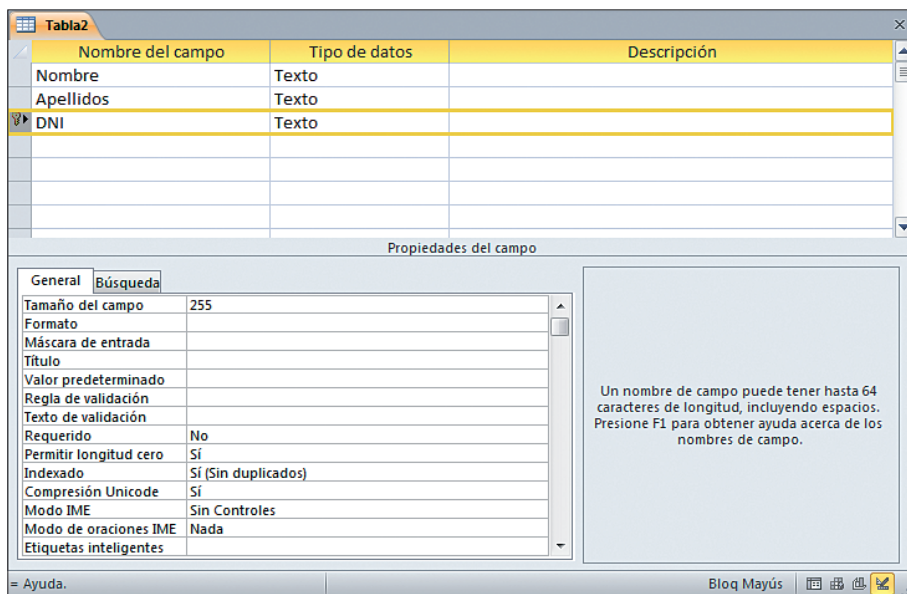
Nombre	Apellidos
José	Álvarez
Jesús	Martín
Juan	Muñoz
Pedro	Muñoz
Antonio	Zambrana
*	

Tabla Indexada (ÍndiceApeNom).

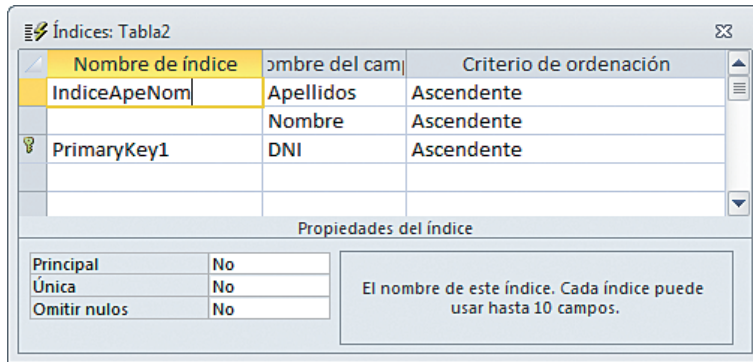
Podremos modificar un índice cuando lo deseemos, por ejemplo eliminando campos, si se trata de un índice de varios campos. Para ello sólo tendremos que pulsar sobre el selector de fila del campo que deseemos eliminar en la ventana de índices y pulsar el botón **Supr**, de la misma forma que hacíamos con los registros en la vista hoja de datos de la tabla. De igual modo podemos eliminar los índices completamente.



En el caso de que deseemos crear un campo clave, sólo tendremos que seleccionar el campo en cuestión en la vista de diseño de la tabla y pulsar el botón **Clave principal** del grupo de opciones **Herramientas**. Una vez hecho esto aparecerá en el selector del campo escogido, en la vista diseño de la tabla, un icono con el dibujo de una llave, que indica que ese campo es la clave principal.



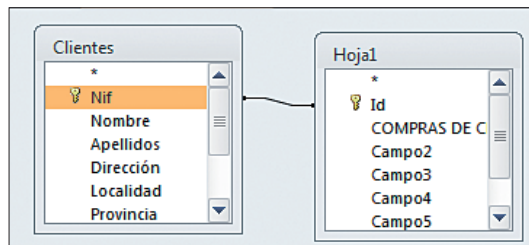
Si después de crear el campo clave, abrimos la ventana de índices, veremos que se ha añadido el campo clave. Observa que por las especiales características del campo clave las opciones de que sea un índice de valor único y de que no permita nulos son obligatorias, además de indicar que es el campo principal.



4. Relación entre Tablas

Se dice que dos tablas están relacionadas entre sí cuando existe un mismo campo, o varios campos iguales, en ambas tablas. Que los campos sean iguales no queremos decir que deban llamarse igual, si no que deben hacer referencia al mismo dato, es decir, podemos tener una tabla **Cientes** con un campo **CódigoPoblación** que indique la población donde reside el cliente, y una tabla **Poblaciones** donde almacenamos todos los datos de las distintas poblaciones (nombre, provincia, código postal,...) que posea un campo llamado **Código** al que **CódigoPoblación** de la tabla **Cientes** hace referencia.

A continuación se puede ver un ejemplo de dos tablas relacionadas:



Se puede ver la línea de unión que representa que dos tablas están relacionadas.

Las relaciones entre tablas se utilizan para la mejor comprensión de la base de datos, creando varias tablas y utilizando cada una ellas para un trabajo concreto. Por ejemplo si vamos a hacer una base de datos en la que tendremos en cuenta los niveles académicos de los alumnos del centro, se usarán dos tablas, una donde estarán definidos los niveles académicos y otra para el alumno; donde se introducirán como campos todos los datos de interés de éste: nombre, apellidos, DNI, dirección, teléfono, etc. y además un campo para relacionarlo con la tabla de niveles académicos donde tendremos un campo clave (código) y la descripción de éste.

El campo CodNvlAcad (Código de Nivel Académico), aparece en ambas tablas, en la tabla “Niveles académicos”, guardaremos como datos todos los niveles académicos, éstos pueden ser, entre otros, los siguientes:

CodNvlAcad	Descripción
1	SIN ESTUDIOS
2	ESTUDIOS PRIMARIOS SIN FINALIZAR
3	CERTIFICADO DE ESTUDIOS PRIMARIOS SIN TÍTULO
4	GRADUADO ESCOLAR
5	FP1 GRADO (TÉCNICO AUXILIAR)
6	FP2 GRADO (TECNICO ESPECIALISTA)
7	TÍTULO DE BACHILLER

De este modo, si en la tabla “Alumnos” introducimos el código de nivel académico 4 le estamos indicando a Access que el alumno tiene el graduado escolar.

4.1. Tipos de Relaciones

Los distintos tipos de relaciones que se pueden crear son:

- **Relación de uno a muchos:** es el tipo de relación más utilizado. En este tipo de relación un elemento de una tabla se relaciona con muchos elementos de la segunda tabla. En el ejemplo anterior, tenemos la tabla “Alumnos” y la tabla “Nivel académico” pues bien, un alumno tiene

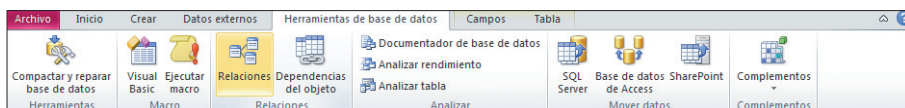
un único nivel académico (siempre elegiremos el nivel académico más alto), y un mismo nivel académico lo pueden tener varios alumnos.

- **Relación de uno a uno:** este tipo de relación, relaciona un elemento de una tabla con otro elemento de la segunda tabla.
- **Relación de muchos a muchos:** en este tipo de relación, varios elementos de una tabla, se relacionan con varios elementos de la segunda tabla. Este tipo de relación la veremos mejor con un ejemplo: si tenemos la tabla “Alumnos” y la tabla “Idiomas”, un alumno puede hablar varios idiomas, y un idioma lo pueden hablar varias personas, por lo que este tipo sería de muchos a muchos. Esta relación es muy difícil de crear, por lo que para hacerlo, en el diseño debemos crear una tabla intermedia que contenga la clave principal de cada una de las tablas a relacionar.

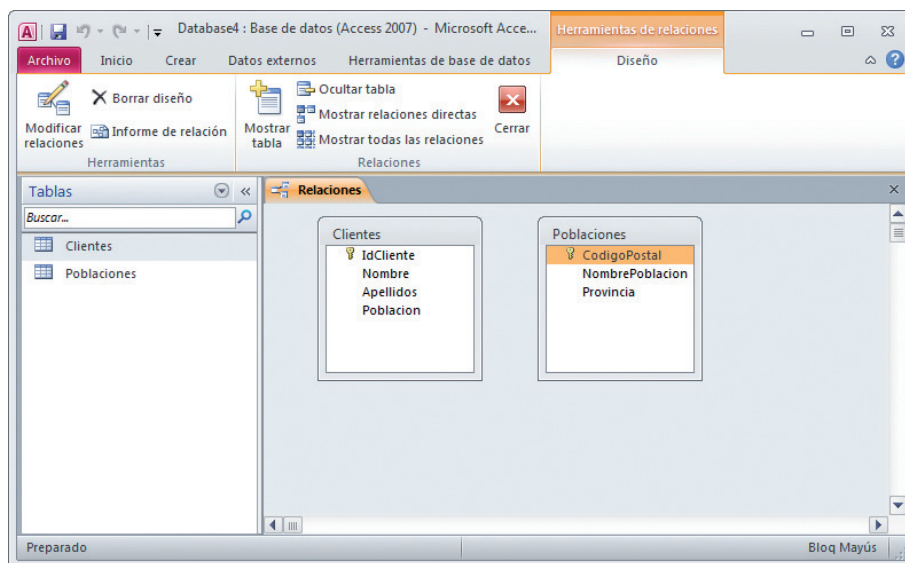
5. Crear una Relación

Para crear una relación entre tablas, éstas deben estar cerradas, ya que no es posible crear ni modificar relaciones entre tablas abiertas.

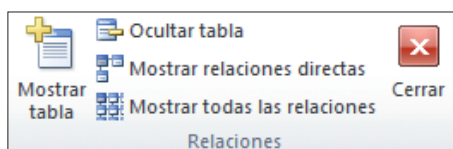
Para crear una relación, debemos situarnos en la cinta de opciones **Herramientas de base de datos**, por lo que debemos situarnos sobre ella. Una vez hecho esto, haremos clic sobre el botón **Relaciones** que se encuentra en el grupo de opciones **Mostrar u Ocultar**:



Una vez hecho esto se abrirá la pestaña **Relaciones** en el área de documento y se abrirá una nueva cinta de opciones llamada **Diseño** con las herramientas necesarias para crear y editar relaciones entre tablas de nuestra base de datos.

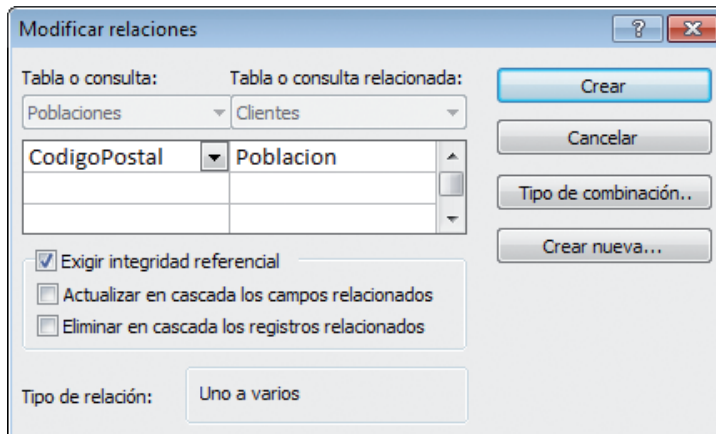


Si la base de datos no tiene ninguna relación definida, se mostrará automáticamente el cuadro de diálogo **Mostrar tabla**. Si necesitamos agregar las tablas que deseamos relacionar y no aparece el cuadro de diálogo **Mostrar tabla**, haremos clic en el botón **Mostrar tabla** del grupo de opciones **Relaciones**.



Si las tablas que deseamos relacionar ya están mostradas, las seleccionaremos. Hecho esto, debemos pulsar sobre el botón **Agregar** y seguidamente pulsar sobre el botón **Cerrar**.

Tras esto, aparecerán las tablas que hemos seleccionado para relacionar con sus correspondientes campos en la pestaña **Relaciones**. Debemos hacer clic sobre el elemento (campo) por el que se hará la relación de la primera de las tablas y sin soltar el botón izquierdo del ratón, arrastrar hasta el elemento a relacionar de la segunda tabla. A continuación aparecerá la siguiente ventana:



Al cerrar esta ventana, *Access* pregunta si deseamos guardar el diseño, independientemente de si se hace o no, *Access* guarda automáticamente las relaciones creadas en la base de datos.

Para que una relación sea del tipo uno a uno, o del tipo uno a muchos, se debemos seleccionar la opción **Exigir integridad referencial**. Dependiendo de las opciones que escojamos, será la relación de un tipo u otro.

6. Integridad Referencial

La integridad referencial es un sistema de reglas que utiliza *Microsoft Access* para garantizar que las relaciones entre los registros de tablas relacionadas sean válidas y que no se eliminan ni modifican accidentalmente datos relacionados.

Se puede establecer la integridad referencial cuando se cumplen las siguientes condiciones:

- Ambas tablas pertenecen a la misma base de datos de Microsoft Access.
- Los campos relacionados tienen el mismo tipo de datos. Aunque existen dos excepciones:
 1. Un campo autonumérico puede estar relacionado con un campo numérico con la propiedad Tamaño del campo establecida a entero largo.
 2. Un campo autonumérico con la propiedad **Tamaño del campo** establecida a **Id. de réplica**, puede estar relacionado con un campo numérico con la propiedad **Tamaño del campo** establecida a **Id. de réplica**.

Cuando exigimos la integridad referencial, deben tenerse en cuenta las siguientes reglas:

1. No puede haber registros en una tabla secundaria, sin que estén enlazados con algún registro de la tabla principal. Por ejemplo, no se puede tener un alumno que esté relacionado con ningún nivel académico.
2. No puede eliminar un registro de una tabla principal si existen registros coincidentes en una tabla secundaria. Por ejemplo, no se puede eliminar un nivel académico si existen alumnos con ese nivel académico asignado.
3. No puede cambiar un valor de clave principal en la tabla principal, si ese registro tiene registros relacionados. Por ejemplo, no se puede cambiar el código de un nivel académico si tiene alumnos relacionados.

Si desea que se cumplan estas normas, debemos seleccionar la casilla **Exigir integridad referencial**.

The screenshot shows a dialog box titled "Modificar relaciones". It contains the following elements:

- Buttons: "Crear", "Cancelar", "Tipo de combinación..", "Crear nueva..."
- Fields: "Tabla o consulta:" (Poblaciones), "Tabla o consulta relacionada:" (Clientes)
- Fields: "CodigoPostal" (selected from a dropdown), "Poblacion" (selected from a dropdown)
- Checkboxes:
 - Exigir integridad referencial
 - Actualizar en cascada los campos relacionados
 - Eliminar en cascada los registros relacionados
- Field: "Tipo de relación:" (Uno a varios)

Si se activa la casilla de verificación Actualizar en cascada los campos relacionados, permite que al modificar un dato de un campo empleado en una relación, varíe en todos los registros de las otras tablas relacionadas.

Si se activa la casilla de verificación Eliminar en cascada los registros relacionados, la eliminación de un registro en la tabla principal, elimina todos los registros relacionados en la tabla relacionada.

Cuando indicamos que una relación mantenga la integridad referencial, ésta se nos mostrará en la ventana de relaciones con una representación que indica el tipo de relación entre las tablas, es decir, si es de uno a muchos o de uno a uno, cosa que antes de exigir la integridad referencial no podía ser asegurada.

