Diseño de Consultas en Microsoft Access*

Una consulta nos permite seleccionar o modificar la información contenida en una Base de Datos. En este sentido, Access distingue entre consultas de selección, mediante las cuales podemos recuperar la información contenida en una o varias tablas, y las consultas de acciones, que permiten añadir nuevos registros, actualizar o eliminar registros ya existentes y crear nuevas tablas.

Además, existen las consultas de Tabla de Referencias Cruzadas que, siendo consultas de selección, permiten mostrar los resultados como tablas de doble entrada. También existen otros tipos de consultas (de Unión, de Paso a Través y de Definición de Datos) que no vamos a considerar en estas notas.

Tabla 1. Tipos de consultas

Tipo de Consulta		**¿Para qué sirve?**
Selección		Recuperar información de una o varias tablas. Se pueden seleccionar únicamente las columnas que nos interesen Se puede seleccionar sólo las filas que cumplan con ciertos criterios Podemos ordenar los resultados según los valores de uno o varios campos Podemos calcular totales (sumas, promedios, …)
Tabla de Referencias Cruzadas		Similares a las consultas de selección, pero muestran la información como una tabla de doble entrada.
Acciones	Datos Anexados	Para agregar información: añadir nuevas filas a una tabla:
	Actualización	Para modificar la información existente
	Eliminación	Eliminar filas de una tabla
	Creación de Tabla	Crear una tabla a partir del resultado de una consulta de selección

Para crear las diferentes consultas, Access nos ofrece un diseñador de consultas visual que permite configurar todos los elementos de las mismas (campos que se quieren mostrar, criterios de selección, criterios de búsqueda, etc.) de una forma completamente intuitiva. La descripción de este diseñador la veremos al hablar de las consultas de selección.

* Apuntes elaborados por Juan Aguarón Joven para la asignatura de Sistemas Decisionales. Universidad de Zaragoza

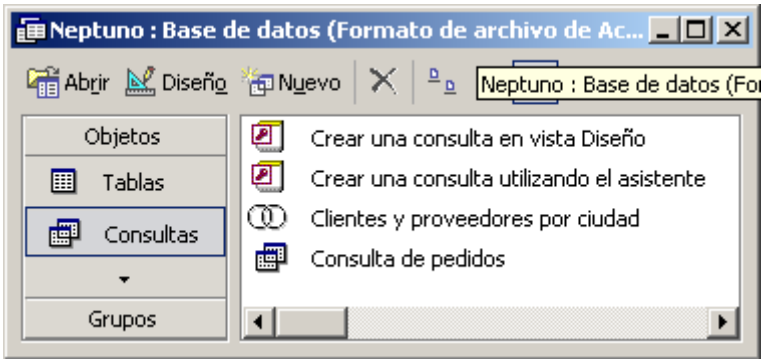
1. Consultas de Selección

Como ya hemos indicado en la Tabla 1, el objetivo de las consultas de selección es recuperar la información almacenada en una o varias tablas de la base de datos. Hay que tener en cuenta que:

- No es necesario incluir todos los campos: podemos seleccionar cuales son la columnas que queremos incluir en la consulta. Por ejemplo, si tenemos una tabla con información completa de nuestros empleados (nombre, apellidos, dirección teléfono, DNI,…) podemos construir una consulta que nos muestre solamente los nombres y apellidos.
- Podemos limitar los registros mediante la inclusión de criterios. Por ejemplo, podemos crear una consulta que nos muestre información **solamente** de aquellos empleados que llevan más de 25 años en la empresa.
- Podemos ordenar los resultados según diferentes criterios. En el ejemplo anterior, es posible ordenar a los empleados por el nombre y apellido. O, si se quiere, por orden de antigüedad.
- Podemos calcular totales. En muchas ocasiones, disponemos de información desagregada y deseamos agruparla para ofrecer resúmenes. Veremos que las consultas de selección nos permiten realizar este proceso. Por ejemplo, disponemos de una tabla que recoge las ventas diarias de los productos de una empresa, pero queremos saber el total de ventas mensuales de un determinado producto.

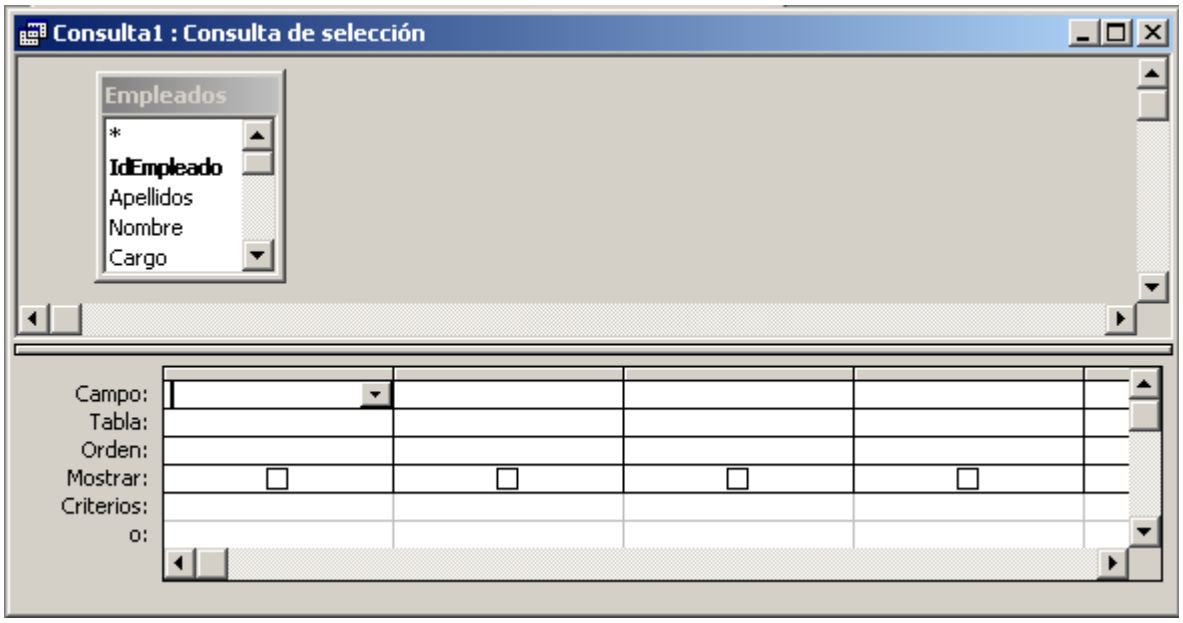
1.1. El diseñador de consultas

Para comenzar a diseñar una consulta nueva debemos seleccionar la opción *Crear una consulta en vista Diseño*, dentro del apartado *Consultas* de la ventana de *Base de Datos*.



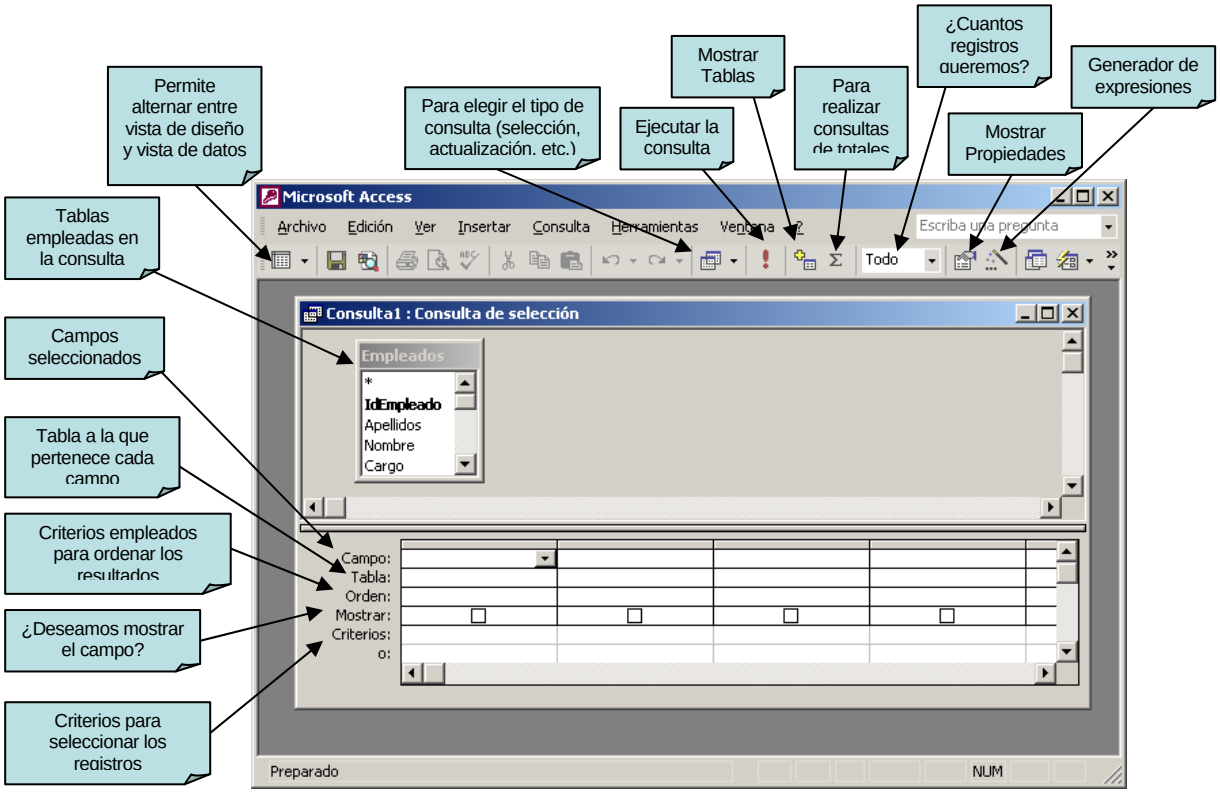
Al tratarse de una consulta nueva, automáticamente nos aparece el diálogo *Mostrar Tabla*, que nos permite seleccionar la tabla o las tablas de las cuales deseamos obtener la información.

Supongamos que deseamos extraer información de la tabla de Empleados. Seleccionamos dicha tabla, pulsamos el botón *Agregar* (o hacemos doble clic sobre el nombre de la tabla) y cerramos el diálogo:



Como vemos, en la parte superior del diseñador nos aparecen las tablas sobre las que queremos trabajar. Si posteriormente deseamos agregar nuevas tablas, podemos volver a mostrar este diálogo anterior pulsando el botón *Mostrar Tablas*  de la barra de herramientas. Si por el contrario deseamos eliminar una tabla del diseñador de consultas, debemos seleccionarla y pulsar la tecla SUPR.

Ahora ya estamos preparados para diseñar la consulta. Sin embargo, vamos a parar un momento para considerar las diferentes opciones que tenemos a la vista en la barra de herramientas y en el diseñador de consultas:



El botón más a la izquierda de la barra de herramientas nos permite seleccionar la forma de ver la consulta. Podemos elegir entre *vista de diseño* (se corresponde con la imagen anterior), *vista de datos* (se muestran los datos seleccionados a partir de la consulta actual) y *vista SQL*, que permite visualizar y modificar la sentencia SQL subyacente. Pulsando el botón se conmuta entre *vista de diseño* y *vista de datos*, siendo necesario recurrir al menú desplegable para seleccionar la *vista SQL*.

Como ya se ha comentado, hay diferentes tipos de consultas. Cuando creamos una nueva consulta ésta es de selección, pero si podemos cambiar el tipo de consulta seleccionándolo del correspondiente botón de la barra de herramientas (también podemos acudir al menú consulta y seleccionar desde ahí el tipo).

Agregar campos

Podemos añadir campos a la consulta de tres maneras diferentes:

- Doble Clic en el campo de la tabla
- Clic + Arrastrar el campo a la columna correspondiente
- Seleccionar el campo del menú desplegable de la fila *Campos* del diseñador

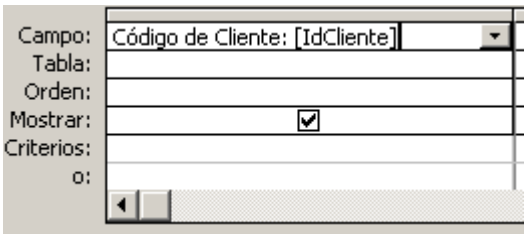
Si queremos agregar todos los campos, doble clic en el campo "*" que aparece al principio de cada tabla (o bien arrastrarlo). Con el *, si modificamos la tabla, no tenemos que modificar la consulta.

No todos los campos que agregamos tienen que verse en la consulta, sólo los que tienen la casilla *Mostrar campo* seleccionada. Por ejemplo podemos emplear campo para realizar una búsqueda pero podemos ocultarlo para que no se muestre en el resultado final.

Si queremos eliminar un campo, hay que seleccionarlo desde el selector de columna (la celda estrecha y gris que se encuentra en la zona superior de cada campo) y pulsar el botón SUPR. Para reorganizar las columnas, se seleccionan de la misma manera y luego se pueden arrastrar desde el mismo selector de columna.

Podemos cambiar el encabezado de los campos sin que ello signifique que se cambia el nombre del campo en la tabla. Esto se conoce como poner un alias a un campo. Para ello se pone en la cuadricula, delante del nombre del campo, el nuevo nombre seguido de ":".

Por ejemplo, si tenemos un campo que se llama "IdCliente" y queremos renombrarlo a "Código de Cliente" podemos escribir: "Código de Cliente: IdCliente".



Ordenando los registros

Para ordenar los registros de una consulta hay que emplear la fila *Orden*, pudiendo indicar para cada campo:

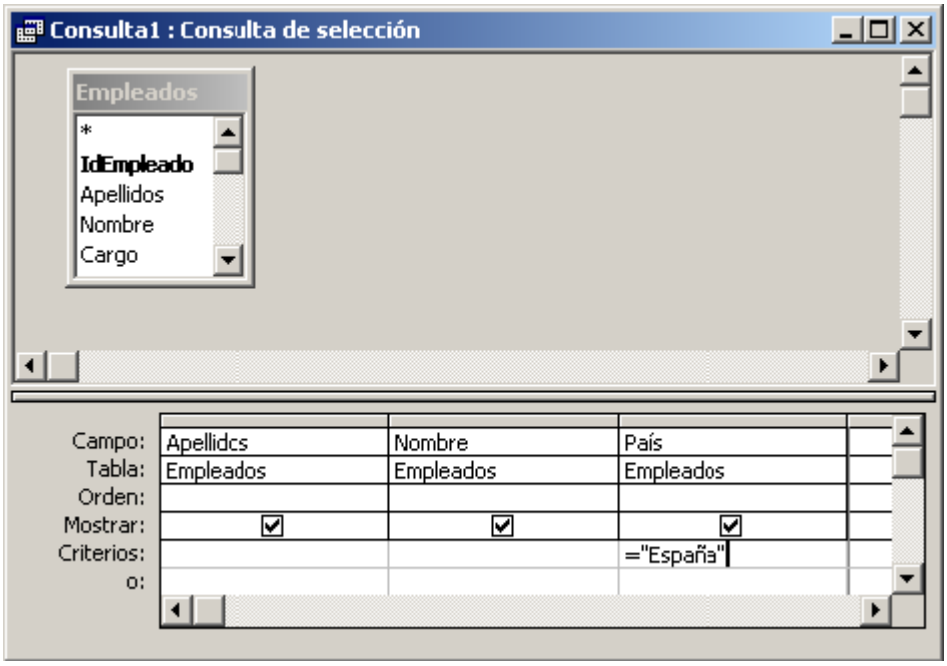
- Ascendente (0-9 y A-Z)
- Descendente (9-0 y Z-A)
- Sin ordenar

Si hay más de un campo empleado para ordenar, el primero que utiliza es el de más a la izquierda y así sucesivamente.

Empleando criterios

En la fila C*riterios* de la cuadricula se pueden especificar los criterios que queremos que cumplan algunos campos (que pueden ser visibles o no al ejecutar los datos de las consultas).

Por ejemplo, queremos mostrar el nombre y apellidos de los empleados cuyo país de origen sea España:



Notar que como el campo País es de tipo texto, tenemos que especificar el criterio de selección poniéndolo entre comillas dobles.

En este caso hemos especificado un criterio de coincidencia exacta (=), pero podemos emplear diversos operadores:

- Igualdad (=): cuando queremos la coincidencia exacta de un campo con el valor especificado. En estos casos la inclusión del símbolo = es optativa.
- Distinto (<>) si queremos que aparezcan todos los registros en los que un campo es distinto al criterio especificado.
- Menor (<) o menor o igual (<=): selecciona sólo los valores que se encuentra por debajo del criterio (en su caso con igualdad)
- Mayor (>) o mayor o igual (>=): selecciona sólo los valores que se encuentra por encima del criterio (en su caso con igualdad)
- Entre dos valores: Permite especificar los valores mínimo y máximo para un campo. Ejemplos: Entre 10 y 20, Entre 1/01/2008 y 31/01/2008.

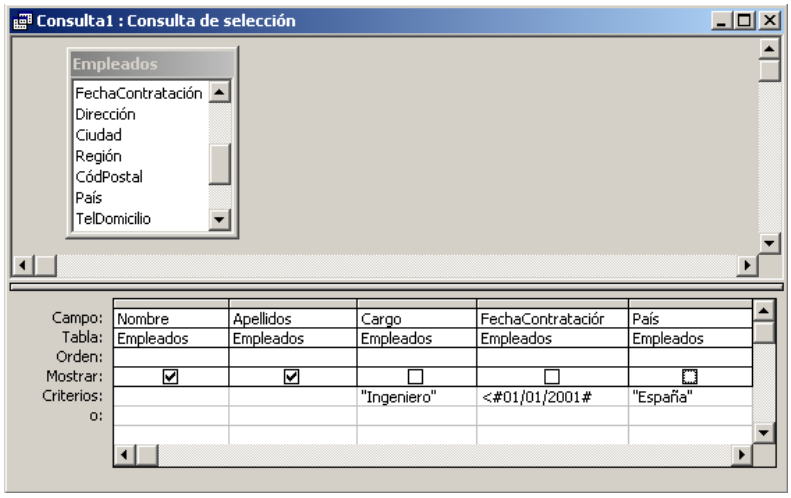
Además, podemos emplear la conjunción *Y* y la disyunción *O* para obtener condiciones más complejas. Por ejemplo, >=1/1/2008 Y <1/07/2008; <=14 O >=65.

Cuando trabajamos con texto, también podemos emplear comodines:

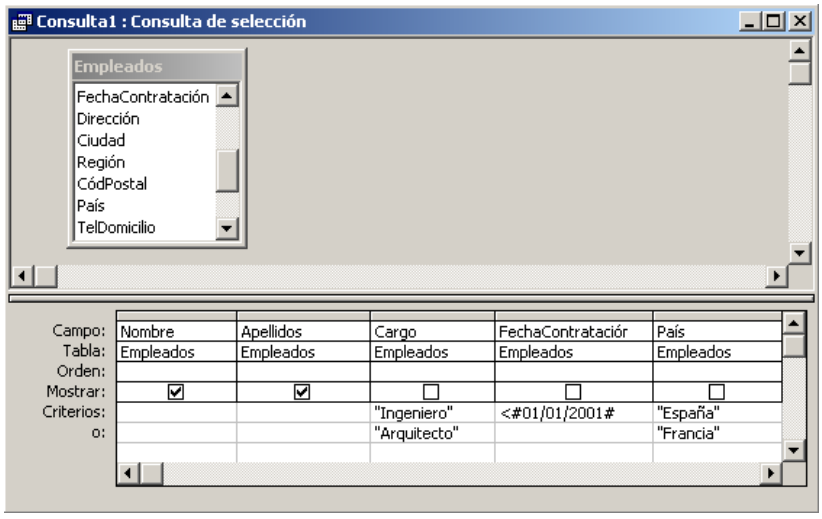
- Un signo de interrogación (?) sustituye a uno y sólo un carácter.
- Un asterisco (*) sustituye a un número cualquiera de caracteres.

A veces es necesario buscar valores nulos o no nulos. Para ello, basta con escribir en el valor del criterio la expresión *Nulo* o *<>Nulo*, respectivamente. Microsoft Access los convertirá automáticamente en las expresiones *Es Nulo* y *Es Negado Nulo* (desafortunada traducción, por cierto, del inglés *is not null*).

Cuando incluimos varias condiciones en la misma fila de criterios, se entiende que se deben cumplir simultáneamente todas ellas. Así, esta consulta se busca el nombre y apellidos de los empleados de la empresa que son ingenieros españoles contratados antes del año 2001.



Si utilizamos filas diferentes de la zona de criterios, se entiende que las condiciones incluidas en cada una de dichas filas son disyuntivas. El resultado final de la consulta está constituido por todos los registros que cumplen con las condiciones de la primera fila, junto con los registros que verifican los de la segunda fila, y así sucesivamente. Por ejemplo, la siguiente consulta nos mostrará los nombres y apellidos de los empleados de la empresa que son ingenieros españoles contratados antes del año 2001 junto con los arquitectos franceses:



Ejercicios propuestos: consultas 1 a 21 de la práctica de Access.

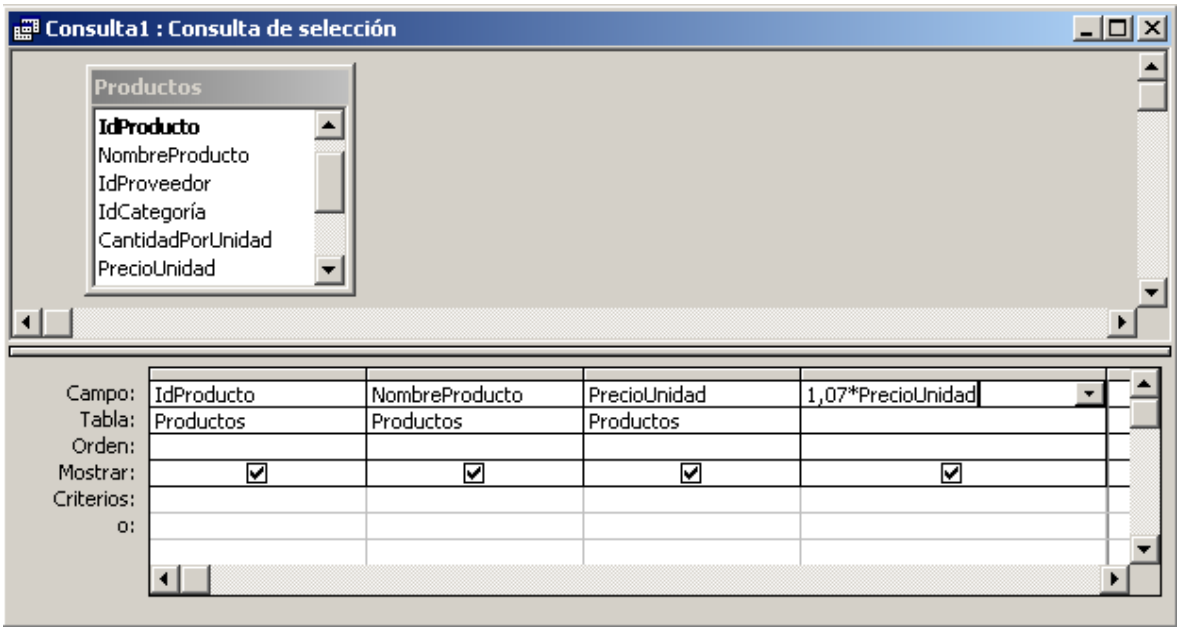
1.2. Campos calculados

Además de mostrar campos existentes en las tablas, las consultas pueden proporcionar nuevas columnas cuyos valores se obtienen a partir de los campos ya existentes. Estas nuevas columnas reciben el nombre de *campos calculados*. El concepto es análogo al empleado en Excel cuando calculamos los valores de una columna a partir de los existentes en las columnas adyacentes. Al igual que ocurre en las hojas de cálculo, para determinar los valores de estas nuevas columnas deberemos realizar operaciones con los datos de partida, por lo que es necesario conocer cuales son las funciones disponibles para ello.

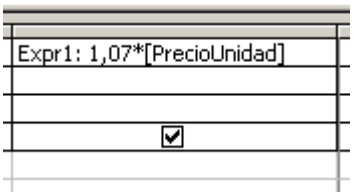
Vamos, en primer lugar, a ver como se construye un campo calculado y más adelante estudiaremos qué funciones tenemos disponibles para ello.

Para obtener un campo calculado basta con escribir en el diseñador de consultas, en la fila *Campo*, una expresión válida que será la que calculará el valor que deseamos obtener. Esta expresión puede contener: valores constantes, operadores, funciones y referencias a otros campos ya existentes.

Por ejemplo, en la tabla *Productos* disponemos de un campo (*PrecioUnidad*) que nos indica el precio de venta sin IVA de los diferentes productos de nuestra empresa. Nosotros deseamos calcular el precio con el IVA del 7% (suponemos por comodidad el mismo tipo de IVA en todos los productos). Lo que necesitamos es que, para cada fila, se muestre el valor *PrecioUnidad* pero incrementado en un 7%, por lo que el nuevo campo responderá a la formula *1.07*PrecioUnidad*. Por lo tanto, esta es la expresión que tendremos que escribir en la fila *Campo* para obtener el nuevo campo calculado:



Al terminar de escribir la expresión, Access la modifica ligeramente, añadiendo un nombre para el nuevo campo (Expr1 en este caso) y delimitando entre corchetes el nombre del campo. Es recomendable dar a estos campos un nombre que se corresponda con la información que contiene. En este caso, podríamos llamarlo *PrecioConIVA* o algo similar.



Como hemos visto, para hacer referencia a un campo en una expresión, basta con escribir el nombre del mismo. Access reconoce dicho campo y lo encierra entre corchetes. Sin embargo, esto no siempre funciona así. Si el campo tiene espacios en blanco en su nombre (por ejemplo, *Precio de Venta*), somos nosotros los que deberemos encerrarlo entre corchetes, puesto que Access no es capaz de delimitar adecuadamente el campo. Es recomendable, siempre que hacemos referencia a un campo, escribirlo entre corchetes para evitar problemas.

Supongamos ahora que el tipo de IVA depende de cada producto de tal manera que dicha información se encuentra almacenada en el campo *TipoIVA* de la tabla de Productos. La consulta que nos daría los precios con IVA sería:

Campo:	IdProducto	NombreProducto	PrecioUnidad	Precio Con IVA:(1+[TipoIVA])*[PrecioUnidad]
Tabla:	Productos	Productos	Productos	
Orden:				
Mostrar:	☑	☑	☑	☑
Criterios:				

En las dos expresiones anteriores estamos empleando valores constantes (el número 1 o el 1.07), operadores (+ y *) y referencias a campos (*TipoIVA* y *PrecioUnidad*). También utilizamos los paréntesis para especificar el orden de las operaciones. Además, podemos emplear los operadores de la resta (-) y la división (/).

Pero si sólo dispusiéramos de estos elementos, la posibilidad de crear campos calculados se vería muy limitada. Para poder crear columnas que nos proporcionen muchas más posibilidades, podemos recurrir, al igual que en Excel, al uso de funciones. Access dispone de un repertorio de funciones bastante amplio que nos permite realizar cálculos más o menos complejos.

Realmente, los operadores de la suma, resta, multiplicación y división no son sino una forma camuflada de acceder a las cuatro funciones matemáticas elementales. Además de éstas, podemos recurrir a funciones matemáticas más complejas (trigonométricas, exponenciales, etc.). Pero también existen funciones que se pueden aplicar para manipular cadenas de texto o para trabajar con fechas. Cuantas más funciones conozcamos, más capaces seremos de crear consultas que puedan proporcionarnos información de utilidad. Además, el uso de funciones puede simplificarnos el diseño de algunas consultas.

1.3. Funciones en Access

En este apartado vamos a describir algunas de las funciones de Access más empleadas al construir consultas. Vamos a distinguir entre funciones que trabajan con valores numéricos, con cadenas de texto o con fechas.

Funciones numéricas

Aparte de las operaciones elementales, disponemos de gran número de funciones matemáticas: Ln(), Sen(), Cos(), Raiz2(), etc. Su uso es similar al de las análogas funciones de Excel.

Funciones para trabajar con cadenas de texto

Estas funciones nos permiten manipular cadenas, obtener información de las mismas o extraer una parte de un texto. Recordemos que, en Access, para hacer referencia a una cadena de texto literal tenemos que encerrarla entre comillas dobles (p.e. "Hola" o "Adiós").[1]

- **Concatenación de cadenas:** para concatenar dos cadenas basta emplear el operador + (también sirve el operador &). Por ejemplo, la expresión "*Sr. "* + *[Apellido1]* proporcionaría un campo nuevo que nos mostraría el apellido de los empleados precedido por el tratamiento *Sr.* (Notar que después de punto hemos dejado un espacio para que no se junte el punto con el apellido).

- **Longitud de una cadena:** Podemos conocer la longitud de una cadena de texto empleando la función *Longitud*.

  Sintaxis: Longitud(cadena)

  Ejemplos: *Longitud("Fernando")* devuelve 8 y *Longitud([Nombre])* devolverá el número de caracteres del nombre de cada empleado.

- **Obtención de parte de una cadena:** Las funciones Izq(), Der() y Medio() nos permiten obtener los caracteres de la izquierda, derecha y medio respectivamente de una cadena de caracteres.

  Sintaxis: Izq(cadena; n) Der(cadena; n) Medio(cadena; inicio; n)

[1] No confundir las dobles comillas con los corchetes. Los corchetes sirven para hacer referencia a un campo mientras que las comillas sirven para delimitar un texto fijo (literal). [Cargo] se refiere a un campo de una tabla cuyo nombre es Cargo, mientras que "Cargo" es una cadena de cinco letras fija. En el primer caso, [Cargo] tomará diferentes valores según el empleado del que se trate: "Arquitecto", "Ingeniero", "Técnico", etc.

Parámetros: cadena es el texto del que se quiere obtener una parte; n es el número de caracteres que queremos obtener; e inicio es la posición del carácter a partir del cual queremos extraer una parte.

Ejemplos: *Izq("Juan"; 1)* devuelve "J"

Der("Pepe"; 2) devuelve "pe"

Medio("ABCDEFG"; 3; 2) devuelve "CD"

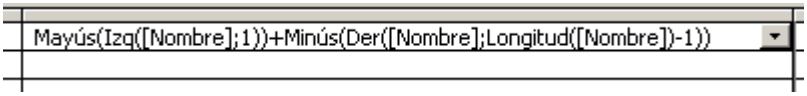
- **Conversión a mayúsculas y minúsculas:** Funciones Mayús() y Minús().

  Sintaxis: Mayús(cadena) Minús(cadena)

  Parámetros: la cadena que se desea convertir

  Ejemplos: *Mayús("Juan")* devuelve "JUAN" *Minús("Pepe")* devuelve "pepe"

Después de haber visto las principales funciones que trabajan con texto, ¿sabrías decir que hace el siguiente campo calculado?



Funciones para trabajar con fechas/horas

Permiten obtener información acerca de la fecha y hora actuales, extraer información de una fecha determinada o realizar cálculos con fechas.

- **Fecha y hora actuales:** Las funciones *Fecha()* y Ah*ora()* devuelven la fecha y la fecha y hora en la que se ejecuta la consulta respectivamente. No requieren ningún parámetro.
- **Extraer información de una fecha:** En su conjunto, una fecha proporciona información acerca del día, del mes y del año[2]. Además, una fecha lleva implícita otra información como puede ser el día de la semana, el trimestre, etc. En muchas situaciones nos interesa acceder a esta información individualizada y para ello tenemos diversas funciones en Access.
  - **Mes(Fecha):** Nos indica el mes al que pertenece una fecha
  - **Año(Fecha):** Obtenemos el año de una fecha

[2] Un campo de tipo fecha puede almacenar también información horaria, según se hayan definido las propiedades del campo. Así, nos podemos encontrar con valores de un campo de tipo fecha como 12/10/2008 13:45:37

- **Día(Fecha):** Extrae de una fecha el día del mes
- **DiaSemana(Fecha):** Devuelve un número entre 1 y 7 indicando el día de la semana
- Las funciones Hora(), Minuto() y Segundo() son análogas y sirven para extraer la hora, minuto o segundo de una fecha/hora.
- La función **ParcFecha()**, DatePart() en inglés, es similar a las anteriores pero es más general ya que permite extraer de una fecha (o fecha/hora) cualquier información asociada a la misma. Es un poco más compleja en su uso (requiere un parámetro adicional), pero es más versátil que las anteriores.

  Sintaxis: ParcFecha(intervalo; Fecha)

  Parámetros: *intervalo* es el intervalo de tiempo que queremos extraer de *Fecha*

  Posibles valores para el parámetro *intervalo*

Valor (español)	Valor (inglés)	Descripción
aaaa	yyyy	Año
t	q	Trimestre
m	m	Mes
a	y	Día del año
d	d	Día
e	w	Día de la semana
ee	ww	Semana
h	h	Hora
n	N	Minutos
s	S	Segundos

  Ejemplos: *DatePart("aaaa"; #10/11/1967#)* devuelve 1967

  DatePart("t"; #10/11/1967#) devuelve 4

  DatePart("d"; #10/11/1967#) devuelve 10

- La función **AgregFecha()**, DateAdd() en inglés, permite hacer cálculos con fechas, agregando a una fecha determinada un intervalo de tiempo. De esta forma podemos contar a partir de una fecha de referencia sumando (o restando) un número de días, semanas, meses, etc.

  Sintaxis: AgregFecha(intervalo; número; Fecha)

  Parámetros: el resultado es incrementar la *Fecha* un *número* de periodos de tiempo (meses, semanas, días,…) indicados por el parámetro *intervalo*. Los

valores del parámetro intervalo son los mismos que se definieron para la función *ParcFecha()*.

Ejemplos: *AgregFecha("yyyy"; 2; #15/06/1994#)* incrementa en dos años (yyyy) la fecha indicada devolviendo #15/06/1996#

AgregFecha("m";-2; #16/05/1996#) decrementa en dos meses la fecha indicada devolviendo #16/03/1996#).

AgregFecha("m"; 1; Fecha()) aumenta en un mes la fecha actual, devolviéndonos la fecha de dentro de un mes.

- Por último, para sumar o restar días a una fecha, es más sencillo sumar o restar directamente el número de días a la fecha.

  Ejemplos: *#25/07/2008# - 10* devuelve la fecha #15/07/2008#

  Fecha() + 5 nos da la fecha de dentro de cinco días

Otras funciones

Existen otras funciones en Access que se emplean para cálculos financieros, para conversión de datos, para tratamiento de errores,… Vamos a ver una función que permite discernir entre dos posibles casos para devolver el valor adecuado.

Al poner un ejemplo para la concatenación de cadenas, hemos considerado una campo calculado con la expresión *"Sr. " + [Apellido1]*. El objetivo era que nos apareciera el tratamiento Sr. antes del apellido de los empleados. Esto está bien siempre y cuando los empleados sean hombres, pero será un error si el empleado es una mujer. Supongamos que tenemos un campo en la tabla de empleados, llamado *Sexo*, que nos indica el sexo ("H" o "M") de cada empleado. Si el sexo de un empleado es "H", deberemos anteponer a su apellido el tratamiento Sr., mientras que si el sexo es "M" deberemos anteponer el tratamiento Sra.

Este tipo de situaciones en las que se pueden presentar dos vías de acción se puede resolver con el empleo de la función *SiInm()*. Su sintaxis es la siguiente:

SiInm(condicion; Valor si Verdadero; Valor Si falso)

donde condición es una expresión (una comparación) que tiene que devolver verdadero o falso. En caso de que sea verdadero, el valor que se obtiene es el indicado por el segundo parámetro, mientras que si es falsa se devuelve el valor indicado por el tercer parámetro.

Ejemplo: SiInm([Sexo]="H"; "Sr."; "Sra.")

Si el empleado es hombre, el campo *Sexo* correspondiente a dicha fila contendrá el valor "H", por lo que la condición se verifica y entonces la función devuelve el valor "Sr.". En caso contrario, devolvería el valor "Sra.".

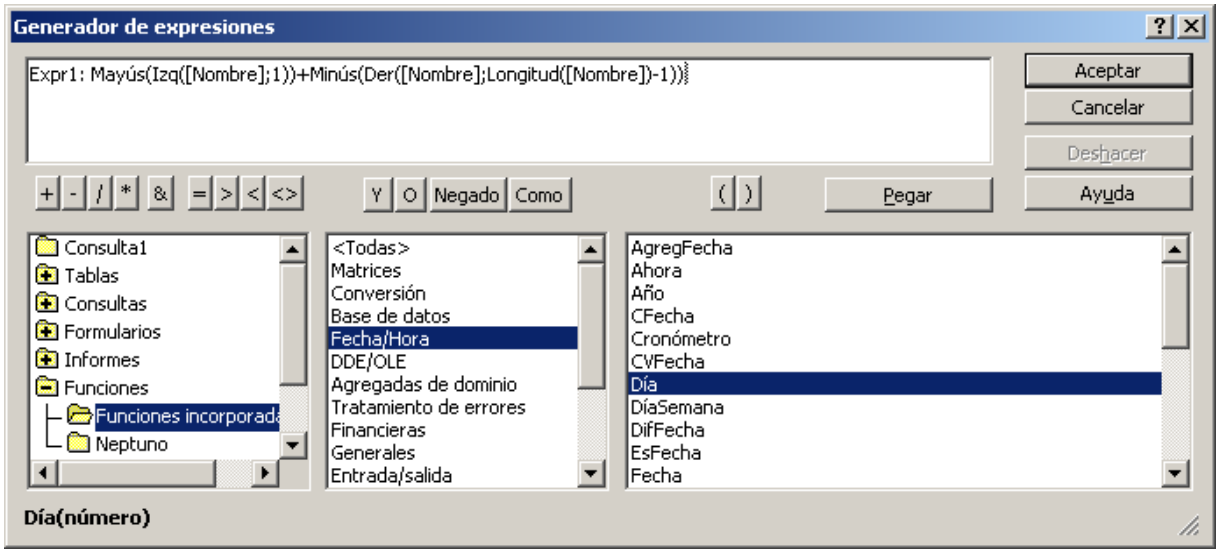
De esta forma podemos construir una expresión más compleja que nos concatene el tratamiento correcto de cada empleado a su apellido:

SiInm([Sexo]="H"; "Sr."; "Sra.") + " " + [Apellido1]

Generador de Expresiones

Al igual que en Excel existe un asistente para escribir funciones, en Access existe un Generador de Expresiones que nos permite construir expresiones seleccionando todos los elementos de forma visual.

Para emplearlo basta con posicionarse en la zona en la que deseamos escribir una expresión y pulsar en el botón Generar:  . En ese momento nos aparece una ventana en la que podemos seleccionar los operadores, campos de las tablas, funciones, etc. que necesitemos.

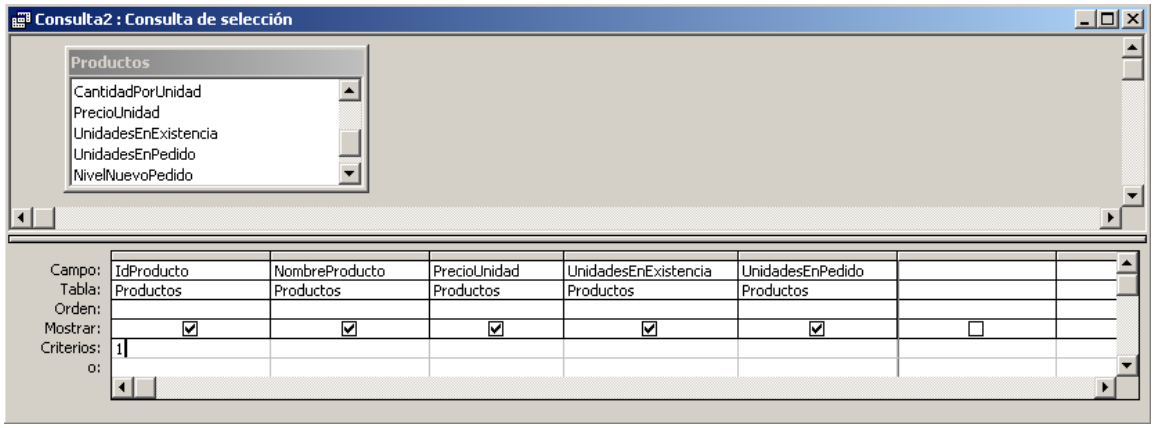


En particular, dentro del apartado Funciones, tenemos el subapartado Funciones incorporadas, donde nos aparecen, agrupadas por categorías, todas las funciones disponibles de Access, con una pequeña ayuda en la parte inferior del diálogo.

También nos permite seleccionar los campos de cualquier tabla de la base de datos.

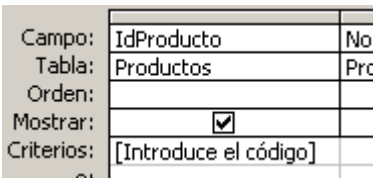
1.4. Consultas con Parámetros

Podemos construir consultas cuyos criterios de selección pueden ser proporcionados cuando la consulta es ejecutada. Supongamos una consulta en la que deseamos obtener información acerca de un producto, para lo cual especificamos como criterios el código de dicho producto, por ejemplo buscamos la información del producto cuyo código es 1:

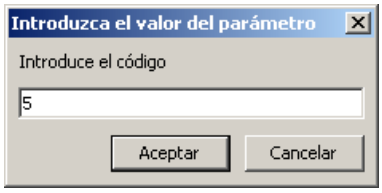


Si ahora deseamos ejecutar la consulta para buscar la información de otro producto, deberemos volver al diseño de la misma, cambiar el valor del criterio y volver a ejecutarla.

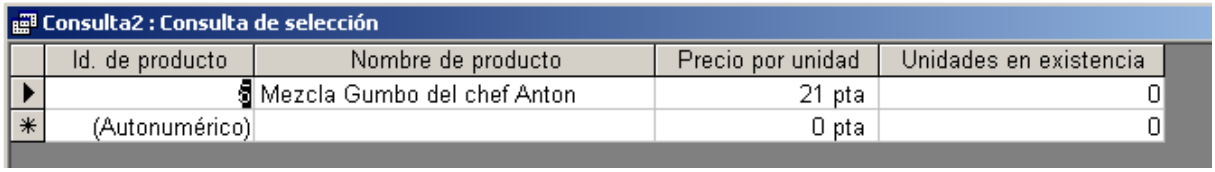
Para hacer las consultas más generales y evitar tener que estar modificando continuamente los criterios, podemos parametrizarla, de forma que el usuario introduzca en el momento de ejecutarse el código del producto que deseamos buscar.



Para ello basta con indicar en la zona de criterios un mensaje entre corchetes. Por ejemplo *[Introduce el código]* o *[Cual es el código del producto]*. Este mensaje es presentado a través de una ventana de información de parámetro donde el usuario debe indicar el código del producto. Una vez introducido, se mostrará la información correspondiente al producto cuyo código ha tecleado.



Al introducir el valor 5, estamos indicando a Access que queremos buscar aquellos productos cuyo código de producto es precisamente este valor. El resultado de la consulta lo vemos en la siguiente salida:

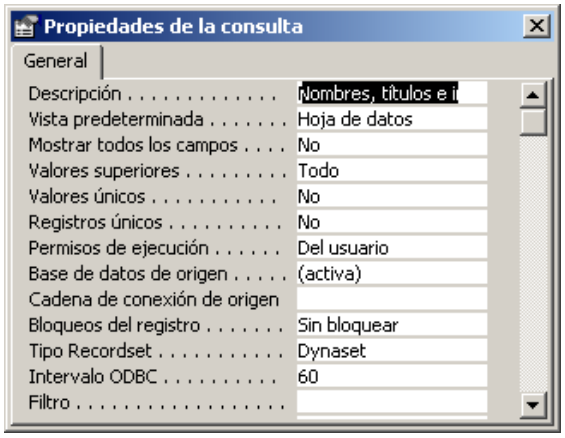


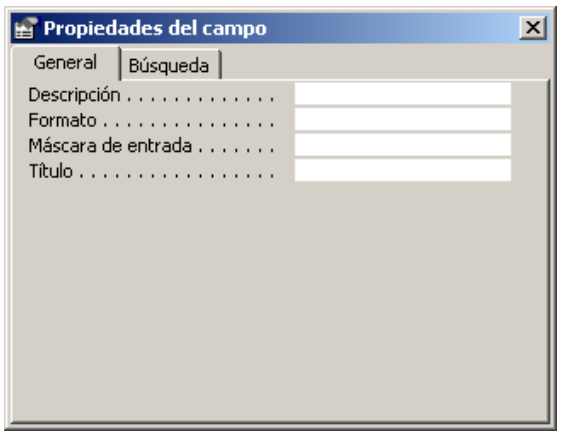
Ejercicios propuestos: consultas 22 a 37 de la práctica de Access.

1.5. Propiedades de las consultas y de los campos

Cuando nos encontramos diseñando una consulta, podemos mostrar la ventana de *Propiedades* pulsando el correspondiente botón de la barra de herramientas:  (también se puede mostrar seleccionando la opción Propiedades del menú contextual).

En ese momento aparece una ventana donde de muestran diferentes propiedades de la consulta. Realmente esta ventana puede mostrar diferente información según el elemento que tenemos seleccionado. Si nos encontramos en la cuadrícula inferior, en la zona correspondiente a un campo, nos mostrará las propiedades de dicho campo, mientras que si hacemos clic en la parte superior del diseñador de consultas, nos aparecen las propiedades generales de la consulta:





Las propiedades del campo afectan al formato de salida del mismo, por lo que no son especialmente relevantes a la hora de diseñar las consultas. Más importantes son las propiedades de la consulta. De todas ellas vamos a comentar algunas:

- Valores superiores: Si indicamos Todo significa que deseamos mostrar todos los registros afectados por la consulta. Si indicamos un porcentaje, p.e, 25%, significa que sólo queremos mostrar el primer 25% de filas. Si indicamos un valor fijo, p.e. 10, estamos indicando que sólo deseamos visualizar las primeras 10 filas.

- La propiedad Valores únicos se puede utilizar cuando se desea omitir registros que contienen datos duplicados en los campos mostrados en la vista Hoja de datos. Supongamos una consulta en la que muestro exclusivamente el proveedor de todos mis productos. Posiblemente habrá proveedores que aparecerán repetidos muchas veces. Si seleccionamos la opción *Valores Único*s a Sí, eliminaremos las repeticiones.

- Si ponemos la propiedad *Registros Únicos* a Sí estamos omitiendo de la consulta registros que en la tabla de partida estén repetidos.

1.6. Trabajando con varias tablas

Normalmente la información que deseamos mostrar se encuentra almacenada en varias tablas. Para que una consulta pueda acceder a esta información es necesario mostrar en el diseñador todas las tablas implicadas. Recordemos que en cualquier momento podemos mostrar nuevas tablas empleando el botón *Mostar Tablas*: 

Si las tablas que empleamos en la consulta tienen alguna relación de integridad referencial entre ellas, dicha relación aparecerá en el diseño de la consulta y nos garantiza que la información que obtengamos de ellas estará relacionada adecuadamente[3].

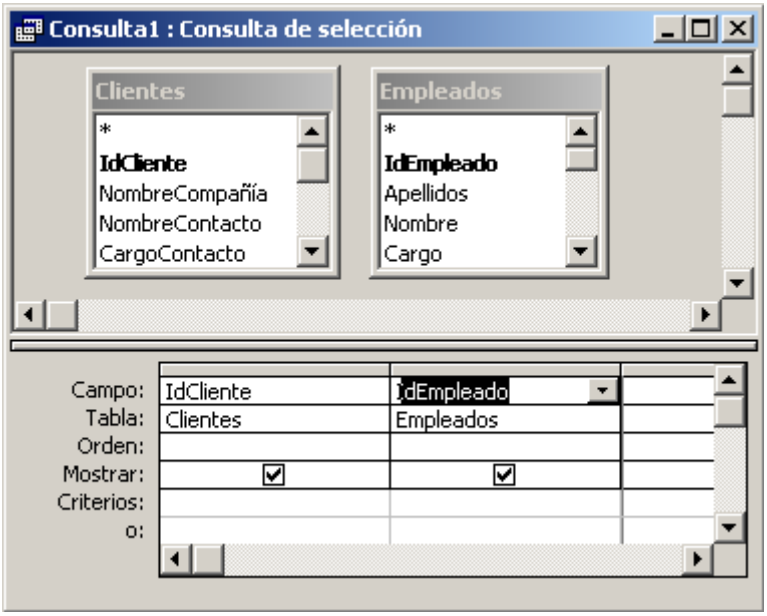
Supongamos que deseamos mostrar la información de los productos de mi empresa pero indicando también el nombre del proveedor. Si analizamos la tabla de *Productos*, vemos que tenemos disponible un campo, *IdProveedor*, que corresponde con el código del proveedor, no con su nombre. Ahora bien, conociendo el código del proveedor, podemos recurrir a la información existente en la tabla de *Proveedores*, buscar el código del proveedor de cada producto y obtener los nombres de los mismos (y al resto de información si fuera necesario).

De esta forma, en esta consulta hay información que se debe obtener de la tabla de *Productos* (código del producto, nombre del producto,…) e información (nombre del proveedor) que se encuentra alojada en la tabla de *Proveedores*. Vamos a incluir ambas tablas en la consulta:

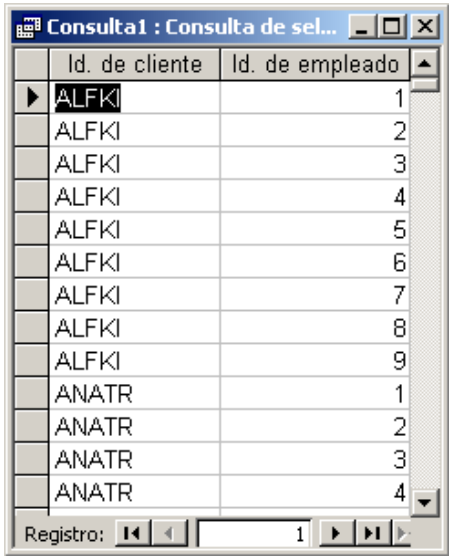
Tal y como hemos comentado, como existe una relación entre las dos tablas, esta se mantiene, indicando que siempre se van a mantener "sincronizadas" las tablas. Si un producto tiene como código de proveedor al número 7, la información que se muestre de la tabla de proveedores será la correspondiente al proveedor número 7.

[3] Si no existe ninguna relación de integridad referencial entre las tablas que empleamos en la consulta, pero queremos relacionarlas en la misma, podemos crea una relación de forma análoga a como se hacía en el menú de Relaciones: arrastrando un campo de una tabla sobre el campo relacionado de la segunda tabla.

Pero, ¿qué pasa si en una consulta incluimos dos tablas que no están relacionadas entre sí? Imaginemos esta consulta:



En este caso no existe ninguna relación entre las tablas de *Clientes* y de *Empleados*, por lo que al seleccionar un campo de cada tabla (código del cliente y código del empleado) se consideran todas las combinaciones de clientes y empleados. En nuestro ejemplo tenemos 91 cliente y 9 empleados, por lo que cada uno de los 91 clientes se empareja con cada uno de los 9 empleados, obteniéndose una consulta con 91 x 9 = ¡819 filas!



En el ejemplo anterior, como la tabla de productos estaba relacionada con la de empleados, cada producto tenía un código de proveedor que enlazaba con la tabla de proveedores, mostrándose exclusivamente la información del único proveedor relacionado. Como en este caso no existe una relación, cada cliente empareja con todos los empleados. A este tipo de combinaciones entre dos tablas se le conoce como *producto cartesiano*. Consiste exclusivamente en la obtención de todos las combinaciones o pares formados por los elemento de ambas tablas.

Muchas veces la obtención todas estas combinaciones es indicativo de que hemos olvidado establecer una relación entre ambas tablas, pero hay situaciones en las que nos interesa formar todas las combinaciones entre dos conjuntos. Por ejemplo, en el caso anterior, nos puede interesar que todos nuestros empleados visiten a nuestros clientes y deseamos precisamente hacer un listado de todos los emparejamientos posibles para ir marcando las visitas realizadas.

En otras ocasiones, puede interesar emparejar tareas con trabajadores para que todos ellos vayan pasando por los diferentes puestos.

Continuando con el mismo ejemplo, vamos a considerar la posibilidad que acabamos de citar, que los empleados de la empresa (vamos a suponer que son comerciales) deben visitar a los clientes para darse a conocer. Eso sí, de todos los emparejamientos posibles (91x9), sólo queremos considerar aquellos en los que tanto el cliente como el empleado residen en la misma ciudad. Es decir, queremos que coincida la ciudad del cliente con la ciudad del empleado. Para ello basta con arrastrar el campo ciudad de la tabla clientes sobre el campo ciudad de la tabla empleados (o viceversa). Automáticamente se crea una relación entre las tablas:

Con ello estamos indicando que sólo queremos que se muestren aquellas combinaciones de clientes-empleados que tienen la misma ciudad.

Nota: La relación que hemos creado es exclusiva de esta consulta. No acabamos de crear ninguna relación de integridad referencial entre las tablas *Clientes* y *Empleados*. Sólo estamos diciendo que en esta consulta deseamos que las dos columnas estén relacionadas. Cuando creemos otra nueva consulta, las dos tablas seguirán estando sin relacionar.

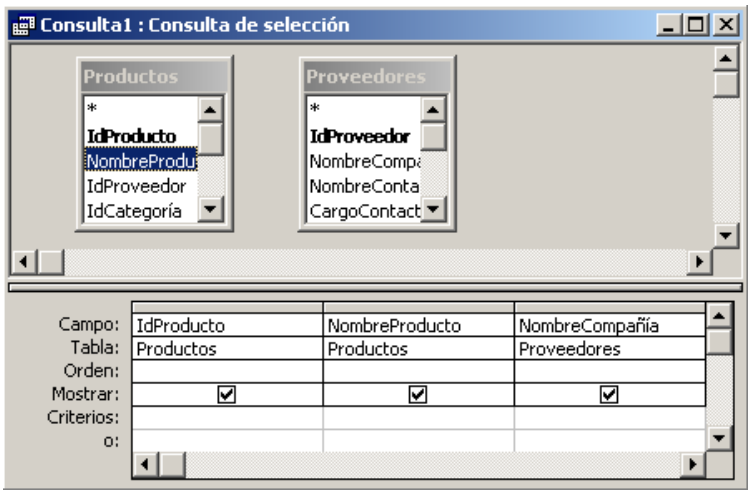
El resultado de la consulta es que sólo se deben realizar 27 visitas. Hemos modificado un poco la consulta para que se muestre información más detallada de los clientes y de los empleados. Vemos que algunos empleados (p.e. Nancy Davolio) sólo deben visitar a un cliente (es el único que se ubica en su ciudad), mientras que otros empleados deben visitar a muchos clientes.

En este último ejemplo hemos partido de dos tablas que no tenían ninguna relación entre sí, pero las hemos relacionado en la consulta porque nos convenía. También se nos puede dar el

caso contrario: que tengamos dos tablas que presenten una relación de integridad referencial pero que no las queramos relacionar en una consulta.

Cuando en una nueva consulta incluimos dos tablas que se encuentren relacionadas (integridad referencial), Access automáticamente considera que deseamos mantener esa relación para la consulta y así se muestra (recordar el ejemplo de productos y proveedores):

Pero supongamos que no deseamos emplear esta relación en nuestra consulta (por el motivo que sea deseamos considerar todas las combinaciones de productos y proveedores). Es tan sencillo como seleccionar la relación (hacer clic sobre la línea que une las dos tablas) y pulsar el botón SUPR:



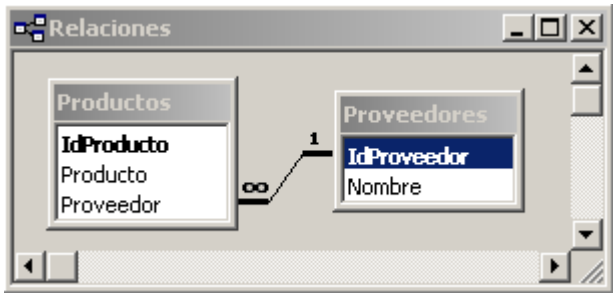
Hay que notar que no hemos eliminado la relación de integridad referencial entre las tablas. Esta se mantiene. Simplemente estamos diciendo que no queremos que las información de los campos IdProveedor de ambas tablas coincida en esta consulta.

Resumiendo, a la hora de realizar consultas podemos relacionar o no dos tablas independientemente de que entre ellas exista una relación de integridad referencial. Eso sí, las relaciones que creemos o eliminemos en una consulta sólo tienen el alcance de la propia consulta.

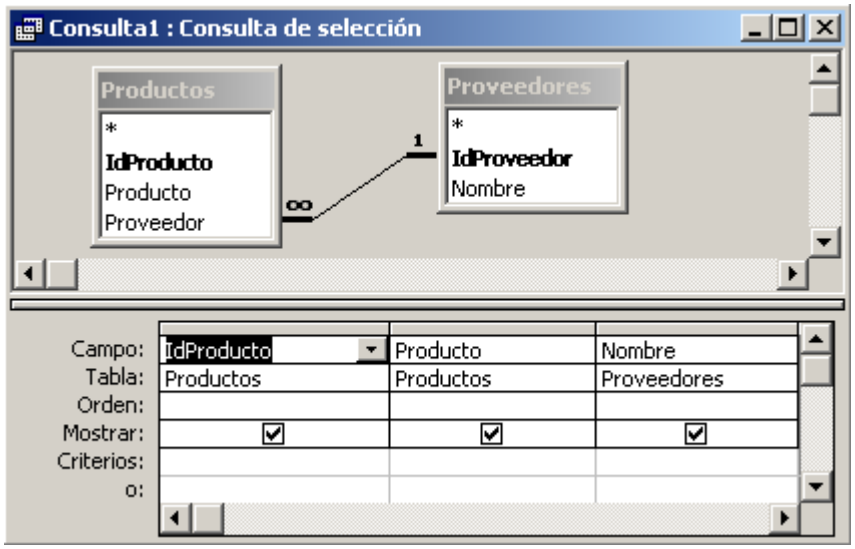
Combinaciones externas

Consideremos ahora el siguiente ejemplo: tenemos una tabla de productos y otra de proveedores. En la tabla *Productos* tenemos los campos *IdProducto* (el código de cada producto), *Producto* (el nombre) y *Proveedor* (código del proveedor habitual de cada producto). La tabla *Proveedores* está constituida por dos columnas: *IdProveedor* (el código del proveedor) y *Nombre*. Disponemos de los siguientes datos:

Además, hemos creado una relación de integridad referencial entre las dos tablas. Con ella exigimos que el proveedor de cada producto exista en la tabla de proveedores de la empresa. Hay que notar que la existencia de una relación de integridad referencial no exige que el campo de la tabla "hija" sea requerido. En nuestro caso podemos ver que el producto 3, Arandelas, no dispone de proveedor: no es necesario rellenar el código de proveedor (eso sí, si lo ponemos debe existir en la tabla *Proveedores).*



A partir de estas tablas vamos a construir una consulta que nos devuelva el listado de todos los productos, incluyendo su código, su nombre y el nombre del proveedor. Al crear una nueva consulta e incluir las tablas Productos y Proveedores obtenemos:



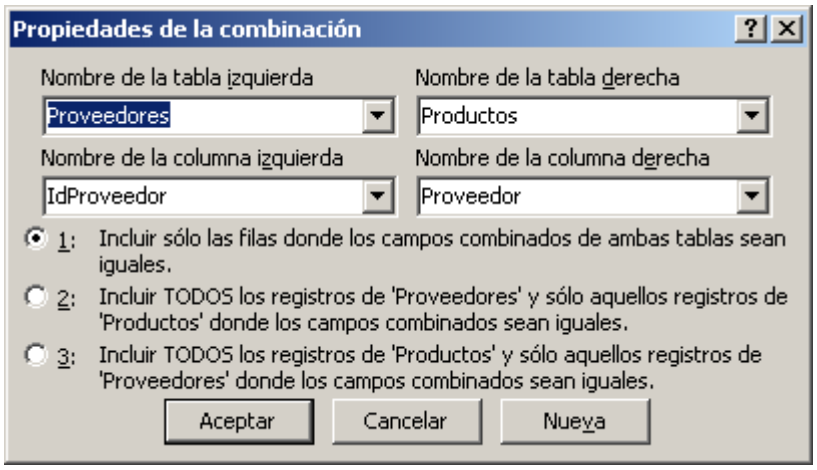
Pero al ejecutar la consulta obtenemos los siguientes registros:

Aparecen tres productos, con sus correspondientes proveedores, pero si nos fijamos en la tabla *Productos* vemos que realmente existen cuatro. En la consulta hemos "perdido" el producto 3, las Arandelas. ¿Qué ha ocurrido? Simplemente que nos muestra el listado de todos los productos que tienen proveedor.

Cuando realizamos una consulta sobre dos tablas y las relacionamos, se entiende que sólo queremos mostrar aquellos registros que tengan coincidentes los campos relacionados. En este caso, como el producto Arandelas tiene nulo el campo proveedor, no existe ninguna fila relacionada en la tabla *Proveedores* y por lo tanto se excluye del resultado final.

Resumiendo: solo se muestran los registros de ambas tablas que tengan iguales los campos relacionados. Los productos que no tienen proveedor no se muestran porque no podemos recupera la información del nombre del proveedor.

Pero en una situación real no nos conformaríamos con esto, es decir, querríamos obtener el listado de todos los productos, indicando, si existe, el nombre del proveedor. Para construir esta consulta, debemos modificar las propiedades de la relación. Si hacemos doble clic sobre la línea que une las dos tablas, obtenemos el siguiente diálogo:



Vemos que, tal y como comentábamos antes, tenemos seleccionada la primera opción, que indica explícitamente que sólo se tendrán en cuenta las filas de ambas tablas que tengan los campos relacionados iguales.

Pues bien, hemos dicho que queremos mostrar todos los productos y, en caso de que exista, el nombre del proveedor. Esto se corresponde con la tercera opción del diálogo. Si la seleccionamos y ejecutamos de nuevo la consulta obtenemos el resultado previsto.

Finalmente, podemos seleccionar la segunda opción y comprobar que es los que ocurre. Vemos que aparecen todos los proveedores emparejados con los productos de los que son proveedores habituales. Si un proveedor no tiene ningún producto asociado, el campo correspondiente será nulo (por ejemplo Drogas Alfonso). Si alguno es proveedor habitual de varios artículos, todos ellos saldrán (Coferdroza):

Ejercicios: Consultas 38 a la 43 de la práctica

1.7. Consultas de agrupamiento