



Las Consultas (I)

Aspectos generales. Tipos de consultas. Criterios y expresiones comunes. Expresiones con parámetros

Jose Ignacio González Gómez

Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna

www.jggomez.eu

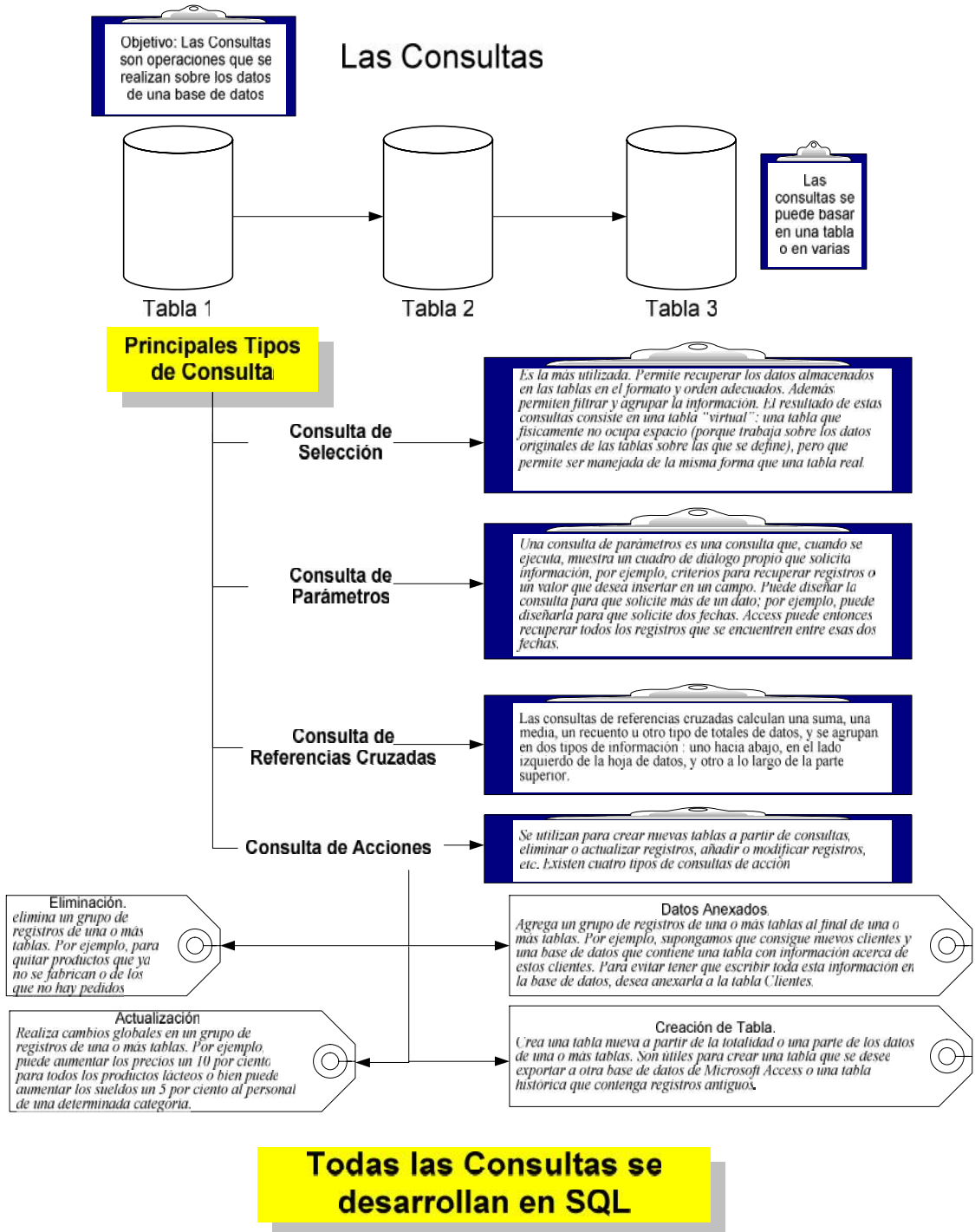
INDICE

1	Introducción a las consultas. Tipos de consultas.....	2
1.1	Aspectos generales.....	2
1.2	Tipos de consultas más frecuentes.....	3
2	Las consultas y el lenguaje SQL.....	5
2.1	Introducción.....	5
2.2	Ejemplo de sentencias SQL en Consultas.....	5
3	Criterios y expresiones comunes en las consultas.....	7
3.1	Consultas con criterios.....	7
3.2	Formando Expresiones. Principales operadores.....	7
3.2.1	Operadores básicos.....	7
3.2.2	Operadores de Condición. Criterios de Búsqueda Y y O.....	8
3.2.3	Operadores de comparación.....	9
3.2.4	Operador Entre.....	9
3.2.5	Operador In.....	10
3.2.6	Operador Es nulo.....	10
3.2.7	Operador Como.....	10
3.3	Expresiones de fecha.....	12
3.4	Expresiones de números, moneda y texto.....	12
3.5	Expresiones lógicas tipo Sí/No.....	12
3.6	Carácter comodín.....	13
4	Expresiones con Parámetros. Empleo de criterios de selección condicionados a la petición del usuario.....	14
5	Bibliografía.....	16

1 Introducción a las consultas. Tipos de consultas

1.1 Aspectos generales

En muchas ocasiones tendremos la necesidad de saber qué clientes viven en tal o cual ciudad, cuantas propiedades del tipo chalet existen, quién paga más de 600 € de alquiler, o cualquier otra pregunta que se nos ocurra. Evidentemente, tenemos el listado completo en forma de tabla, pero según el tipo de preguntas que queramos realizar a la BD nos puede llevar bastante tiempo localizar algún dato en concreto.



Para ello están las consultas. Las consultas con preguntas preparadas que podemos diseñar para utilizar cuando sea necesario. Casi todo el trabajo en Access se basa en las consultas. Si queremos hacer un informe de unos datos determinados, podemos hacerlo a partir de una consulta. Si queremos un listado por pantalla o por impresora, lo haremos de una consulta.

1.2 Tipos de consultas más frecuentes

Por tanto las consultas son operaciones que se realizan sobre los datos de una base de datos. Estas operaciones pueden ser de diversos tipos:

- **Consulta de selección: es la más utilizada.** Permite recuperar los datos almacenados en las tablas en el formato y orden adecuados. Además permiten filtrar y agrupar la información. El resultado de estas consultas consiste en una tabla “virtual”: una tabla que físicamente no ocupa espacio (porque trabaja sobre los datos originales de las tablas sobre las que se define), pero que permite ser manejada de la misma forma que una tabla real.
- **Consulta de Parámetros.** Una consulta de parámetros es una consulta que, cuando se ejecuta, muestra un cuadro de diálogo propio que solicita información, por ejemplo, criterios para recuperar registros o un valor que desea insertar en un campo. Puede diseñar la consulta para que solicite más de un dato; por ejemplo, puede diseñarla para que solicite dos fechas. Access puede entonces recuperar todos los registros que se encuentren entre esas dos fechas. Las consultas de parámetros también son útiles cuando se emplean como base para formularios, informes y páginas de acceso a datos. Por ejemplo, puede crear un informe de ingresos mensuales basado en una consulta de parámetros. Al imprimir el informe, Access muestra un cuadro de diálogo que solicita el mes para el que se desea obtener el informe. Cuando se especifica un mes, Access imprime el informe correspondiente.
- **Consultas de tablas de referencia cruzadas:** presenta los datos en forma de filas y columnas. Es decir las consultas de referencias cruzadas se utilizan para calcular y reestructurar datos de manera que su análisis sea más sencillo. Las consultas de referencias cruzadas calculan una suma, una media, un recuento u otro tipo de totales de datos, y se agrupan en dos tipos de información: uno hacia abajo, en el lado izquierdo de la hoja de datos, y otro a lo largo de la parte superior.
- **Consultas de acciones:** se utilizan para crear nuevas tablas a partir de consultas, eliminar o actualizar registros, añadir o modificar registros, etc. Por tanto una consulta de acción es una consulta que realiza cambios o desplazamientos de muchos registros en una sola operación. Existen cuatro tipos de consultas de acción: de eliminación, de actualización, de datos anexados y de creación de tabla.
 - **Consulta de eliminación.** Una consulta de eliminación elimina un grupo de registros de una o más tablas. Por ejemplo, puede utilizar una consulta de eliminación para quitar productos que ya no se fabrican o de los que no hay pedidos. Con las consultas de eliminación, siempre se eliminan registros enteros, no sólo campos seleccionados dentro de los registros.
 - **Consulta de actualización** Realiza cambios globales en un grupo de registros de una o más tablas. Por ejemplo, puede aumentar los precios un 10 por ciento para todos los productos lácteos o bien puede aumentar los sueldos un 5 por ciento al personal de una determinada categoría. Con una consulta de actualización, puede cambiar los datos de las tablas existentes.

- Consulta de datos anexados *Agrega un grupo de registros de una o más tablas al final de una o más tablas. Por ejemplo, supongamos que consigue nuevos clientes y una base de datos que contiene una tabla con información acerca de estos clientes. Para evitar tener que escribir toda esta información en la base de datos, desea anexarla a la tabla Clientes.*
- Consulta de creación de tabla *Crea una tabla nueva a partir de la totalidad o una parte de los datos de una o más tablas. Las consultas de creación de tabla son útiles para crear una tabla que se desee exportar a otra base de datos de Microsoft Access o una tabla histórica que contenga registros antiguos*

2 Las consultas y el lenguaje SQL

2.1 Introducción

Las consultas a los datos de las bases de datos se hacen a través de los denominados “lenguajes de consulta”. El más utilizado de este tipo de lenguajes es el SQL (Standard Query Language). Access permite realizar consultas en SQL. Pero, dada la filosofía visual seguida en Access, también se permite la definición de consultas a través de una interfaz visual. Ambos métodos son compatibles e intercambiables: se puede pasar de uno a otro sin problemas. Sin embargo, SQL es un estándar que aparece en multitud de sistemas sin variaciones, mientras que el desarrollo visual depende del fabricante.

Una consulta SQL está compuesta por una instrucción SQL que define esa consulta. Se trata de un comando que puede ocupar cuantas líneas de texto se desee, terminado en punto y coma (;). SQL es un lenguaje sencillo e intuitivo: las consultas se asemejan al lenguaje natural.

Existen algunas palabras reservadas, como en cualquier lenguaje: SELECT, INSERT, DELETE, UPDATE, SET, WHERE, IN, DISTICT, GROUP, ORDER, BY, etc.

2.2 Ejemplo de sentencias SQL en Consultas

La consulta más simple posible consiste en la selección de campos y registros de una tabla. Se identifican los campos que nos interesan y una condición que deben cumplir los registros seleccionados. El resultado es una tabla que es un subconjunto de la original.

El formato genérico de este tipo de consultas es:

```
SELECT <lista de campos> FROM <tabla> WHERE <condición>;
```

Esta instrucción recupera ciertos campos de los registros de una tabla que verifican una condición. La cláusula WHERE es opcional. Si se omite, se seleccionan todos los registros (se supone que la condición es siempre verdadera).

```
SELECT <lista de campos> FROM <tabla>;
```

Si nos interesan todos los campos podemos utilizar el símbolo * para identificar a la lista completa:

```
SELECT * FROM <tabla> WHERE <condición>;
```

Si no, podemos especificar varios campos identificándolos por sus nombres y separándolos por comas (,).

```
SELECT campo1, campo2, ..., campoN FROM <tabla> WHERE <condición>;
```

Supongamos la tabla de Alumnos

Campo	Descripción
NIF	NIF
Nombre	Nombre completo
Apellido1	Primer apellido
Apellido2	Segundo apellido
Edad	Edad
Parcial1	Nota obtenida en el primer parcial
Parcial2	Nota obtenida en el segundo parcial
Prácticas	Nota obtenida en las prácticas

Podemos definir las siguientes consultas de ejemplo:

Consulta	SQL
Seleccionar los alumnos mayores de 25 años.	SELECT * FROM Alumnos WHERE Edad>=25;
Seleccionar los apellidos y nombre de los que han aprobado los dos parciales.	SELECT Apellido1, Apellido2, Nombre FROM Alumnos WHERE (Parcial1>=5) AND (Parcial2>=5);
Generar un listado con los nombres y apellidos de los alumnos y sus notas.	SELECT Nombre, Apellido1, Apellido2, Parcial1, Parcial2, Prácticas FROM Alumnos;

Es posible consultar, desde una base de datos, una tabla que pertenezca a otra base de datos. En este caso utilizaremos la sintaxis:

```
SELECT <lista de campos> FROM <tabla> IN <base de datos>;
```

La cláusula IN permite especificar otra base de datos como origen de la tabla.

Ejemplo:

Consulta	SQL
Seleccionar los alumnos mayores de 25 años, suponiendo que la tabla <i>Alumnos</i> está en otra base de datos que se llama <i>C:\COLEGIO\REGITSRO.MDB</i> .	SELECT * FROM Alumnos IN "C:\COLEGIO\REGITSRO.MDB" WHERE Edad>=25;

Esta estructura permite también acceder a datos que se encuentren almacenados en otras bases de datos que no sean Microsoft Access, siempre y cuando Access se encuentre correctamente instalado y configurado.

3 Criterios y expresiones comunes en las consultas

3.1 Consultas con criterios

Una de las mayores ventajas de las consultas es la de introducir criterios de consulta. Mediante los criterios podemos seleccionar registros que cumplan unas condiciones determinadas. Veamos cómo crear una consulta que nos muestre datos sobre los clientes que pagan más de 85.000 um de alquiler mensual.

- Crea una nueva consulta de la tabla CLIENTES y carga los campos Código de propiedad, Nombre y Alquiler mensual

- En la casilla Criterios del campo Alquiler mensual, escribe: >85000

Campo:	Código propiedad	Nombre	Alquiler mensual
Tabla:	CLIENTES	CLIENTES	CLIENTES
Orden:			
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:			>85000
o:			

- Visualiza el resultado de la consulta con el botón Vista

Observa que aparecen los tres campos de los clientes que pagan más de 85.000 um al mes.

- Vuelve al modo Diseño desde el mismo botón

- Agrega el campo Oficina vendedora y añade los siguientes criterios:

Campo:	Código propiedad	Nombre	Alquiler mensual	Oficina vendedora
Tabla:	CLIENTES	CLIENTES	CLIENTES	CLIENTES
Orden:				
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:			>85000	"Terrassa"
o:				

- Visualiza el resultado

En este último caso tenemos dos condiciones que se tienen que cumplir. Sólo veremos los campos de los registros que cumplan las dos condiciones: que paguen más de 85.000 um y cuya oficina vendedora esté en la ciudad de Terrassa.

3.2 Formando Expresiones. Principales operadores

Una expresión se forma combinando un operador con uno o generalmente dos operandos. La mayoría de las veces un operando será un campo de la tabla y el otro un valor concreto u otro campo. Un operando puede ser un nombre de columna, una expresión, un valor concreto o una función predefinida.

3.2.1 Operadores básicos

- +, se utiliza para sumar dos números.
- -, se utiliza para hallar la diferencia entre dos números.
- *, se utiliza para multiplicar dos números.
- ^, se utiliza para elevar un número a la potencia del exponente (número ^ exponente)
- / , se utiliza para dividir dos números y obtener un resultado de signo flotante.
- \, se utiliza para dividir dos números y obtener un resultado entero.

- Mod, divide dos números y devuelve sólo el resto.
- &, se utiliza para concatenar dos expresiones de tipo texto.

También se puede utilizar el operador de suma + cuando los dos operandos son de tipo texto, para concatenarlos. Mi consejo es utilizar el operador & para la concatenación y así evitar errores o confusiones.

3.2.2 Operadores de Condición. Criterios de Búsqueda Y y O

Para seleccionar filas tenemos que **indicar un criterio de búsqueda**, un criterio de búsqueda es una condición que deberán cumplir todas las filas que aparezcan en el resultado de la consulta.

Normalmente la condición estará basada en un campo de la tabla por ejemplo para seleccionar los alumnos de Valencia la condición sería población = "**Valencia**". Para escribir esta condición en la cuadrícula tenemos que tener en una de las columnas de la cuadrícula el campo población y en esa columna ponemos en la fila **Criterios:** el resto de la condición o sea = "**Valencia**".

Cuando la condición es una igualdad no es necesario poner el signo =, podemos poner directamente el valor **Valencia** en la fila **Criterios:** ya que si no ponemos operador asume por defecto el =.

Para indicar **varias condiciones** se emplean los operadores **Y** y **O**.

En un criterio de búsqueda en el que las condiciones están unidas por el operador **Y**, para que el registro aparezca se deben cumplir todas las condiciones. Por ejemplo **precio > 100 y precio < 1200**, aparecen los registros cuyo precio está comprendido entre 101 y 1199.

En un criterio de búsqueda en el que las condiciones están unidas por el operador **O**, el registro aparecerá en el resultado de la consulta si cumple al menos una de las condiciones.

Todas las condiciones establecidas en la **misma fila** de la cuadrícula quedan unidas por el operador **Y**.

En el ejemplo siguiente serían alumnos de Valencia Y cuya fecha de nacimiento esté comprendida entre el 1/1/60 y el 31/12/69.

Campo:	Apellidos Alumnado	Nombre Alumnado	Población	Fecha de nacimiento
Tabla:	Alumnado	Alumnado	Alumnado	Alumnado
Orden:	Ascendente			
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:			"Valencia"	Entre #01/01/1960# Y "\$04/06/62#
o:				

Visualizaremos de la tabla Alumnado los campos Apellidos, Nombre, Población y Fecha nacimiento, los alumnos aparecerán ordenados por Apellidos pero

únicamente aparecerán aquellos que sean de Valencia y hayan nacido entre el 1/1/60 y el 31/12/69, o bien aquellos de Alicante sea cual sea su fecha de nacimiento.

¡Ojo! El criterio de la fecha de nacimiento únicamente afecta a la población Valencia por encontrarse en la misma fila.

Access no diferencia entre mayúsculas y minúsculas a la hora de realizar la búsqueda de registros.

Presentamos a continuación los siguientes operadores que podemos utilizar en una condición.

3.2.3 Operadores de comparación

Estos operadores comparan el valor de una expresión con el valor de otra. Independientemente del operador si uno de los valores es nulo, el resultado de la comparación será nulo (ni verdadero ni falso). Los operadores de comparación que podemos utilizar son:

Operador	Significado
=	igual que
<>	distinto de
<	menor que
<=	menor o igual
>	mayor que
>=	mayor o igual

3.2.4 Operador Entre

Tiene la siguiente sintaxis: Expresión Entre valor1 Y valor2

Examina si el valor de la expresión está comprendido entre los dos valores definidos por valor1 y valor2. Normalmente la expresión será un nombre de campo.

Ejemplo:

[fecha de nacimiento] entre #01/01/60# y #04/06/62#, en la cuadrícula QBE se pondría:

Campo:	Apellidos Alumnado	Nombre Alumnado	Población	Fecha de nacimiento
Tabla:	Alumnado	Alumnado	Alumnado	Alumnado
Orden:	Ascendente			
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:			"Valencia"	Entre #01/01/1960# Y "#04/06/62#"
o:				

3.2.5 Operador In

Tiene la siguiente sintaxis: Expresión In (valor1, valor2, ...)

Examina si el valor de la expresión es uno de los valores incluidos en la lista de valores escritos entre paréntesis.

Por ejemplo, para seleccionar los alumnos de Alicante, Elche, Elda y Onda podríamos poner la condición Población In (Alicante, Elche, Elda, Onda)

Campo:	Nombre Alumnado	Población
Tabla:	Alumnado	Alumnado
Orden:		
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:		In ('Alicante','Elche','Elda','Onda')

3.2.6 Operador Es nulo

Cuando una columna que interviene en una condición contiene el valor nulo, el resultado de la condición no es verdadero ni falso, sino nulo, sea cual sea el test que se haya utilizado. Por eso si queremos listar las filas que no tienen valor en una determinada columna, no podemos utilizar la condición columna = nulo debemos utilizar un operador especial, el operador Es nulo.

Tiene la siguiente sintaxis:

Expresión Es nulo, donde expresión normalmente será un nombre de columna.

Por ejemplo queremos saber los alumnos que no tienen población, la condición sería población Es Nulo

Campo:	Nombre Alumnado	Población
Tabla:	Alumnado	Alumnado
Orden:		
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:		Es Nulo

3.2.7 Operador Como

Se utiliza cuando queremos utilizar caracteres comodines para formar el valor con el que comparar. Por ejemplo queremos visualizar los alumnos cuyo nombre acabe en 'o'. En este caso hay que utilizar el comodín * y utilizar el operador Como para que Access reconozca el * como un comodín y no como el carácter asterisco. La condición sería nombre como '*o'.

El valor que contiene los comodines se conoce como patrón y tiene que ir encerrado entre comillas (simples o dobles). La sintaxis es la siguiente:

En la siguiente tabla mostramos los caracteres comodines que se pueden poner en un patrón y su significado.

Caracteres en patrón	Significado
?	Un carácter cualquiera
*	Cero o más caracteres
#	Un dígito cualquiera (0-9)
[listacaracteres]	Un carácter cualquiera de listacaracteres
[!listacaracteres]	Un carácter cualquiera no incluido en listacaracteres

Listacaracteres representa una lista de caracteres y puede incluir casi cualquier carácter, incluyendo dígitos, los caracteres se escriben uno detrás de otro sin espacios en blanco ni comas. Por ejemplo para sacar los nombres que empiezan por a,g,r o v el patrón sería: '[agrv]*'

Los caracteres especiales corchete de apertura [, interrogación ?, almohadilla # y asterisco * dejan de ser considerados comodines cuando van entre corchetes. Por ejemplo para buscar los nombres que contienen un asterisco, el patrón sería: '*[*]*' en este caso el segundo * dentro del patrón no actúa como comodín sino como un carácter cualquiera porque va dentro de los corchetes.

Si no se encierra entre corchetes, la exclamación ! representa el carácter exclamación.

El corchete de cierre] se puede utilizar fuera de una listacaracteres como carácter independiente pero no se puede utilizar en una listacaracteres. Por ejemplo, el patrón 'a]*' permite encontrar nombres que empiecen por una a seguida de un corchete de cierre.

La secuencia de caracteres [] se considera una cadena de caracteres de longitud cero ("").

Se puede especificar un intervalo de caracteres en listacaracteres colocando un guión - para separar los límites inferior y superior del intervalo.

Por ejemplo, la secuencia [A-Z] en patrón representa cualquier carácter comprendido en el intervalo de la A a la Z.

Cuando se especifica un intervalo de caracteres, éstos deben aparecer en orden ascendente (de menor a mayor).[A-Z] es un intervalo válido, pero [Z-A] no lo es.

Se pueden incluir múltiples intervalos entre corchetes, sin necesidad de delimitadores.

El guión - define un intervalo únicamente cuando aparece dentro de los corchetes entre dos caracteres, en cualquier otro caso representa el carácter guión.

Por ejemplo queremos saber los alumnos de la provincia de Valencia (son los que tienen un código postal que empieza por 46 seguido de tres dígitos cualesquiera, la condición podría ser [código postal] como '46###'

Campo:	Nombre Alumnado	Código Postal
Tabla:	Alumnado	Alumnado
Orden:		
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:		Como '46###'

Campo:	Nombre Alumnado	Código Postal
Tabla:	Alumnado	Alumnado
Orden:		
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criterios:		Como '46###'

En las consultas que hemos hecho, hemos introducido expresiones para los criterios muy sencillas. Vamos a ver qué tipo de expresiones avanzadas podemos utilizar para los criterios.

3.3 *Expresiones de fecha*

Podemos utilizar operadores de comparación típicos de programas informáticos como mayor que, menor que, igual, distinto, etc. Para la cuestión de fechas, podemos utilizar esos mismos operadores. Por ejemplo, imaginemos que queremos saber los registros con un campo cuya fecha se menor al 12 de Junio del 97. Podemos hacerlo de las siguientes formas:

<12 Junio 1997
<#12/06/97#
<12 Jun 97
<12-Jun-97
<12/06/97

3.4 *Expresiones de números, moneda y texto*

Cuando utilicemos campos de este tipo, tan sólo tendremos que escribir los datos y si contienen decimales, el separador de decimales.

En caso de expresión de texto, bastara con escribir el texto utilizando mayúsculas o minúsculas indistintamente.

3.5 *Expresiones lógicas tipo Sí/No*

Para valores tipo Sí podemos escribir las siguientes expresiones en la casilla de criterios:

"Sí"
"Verdadero"
"Activado"
"-1"

Para valores del tipo No podemos usar las siguientes expresiones:

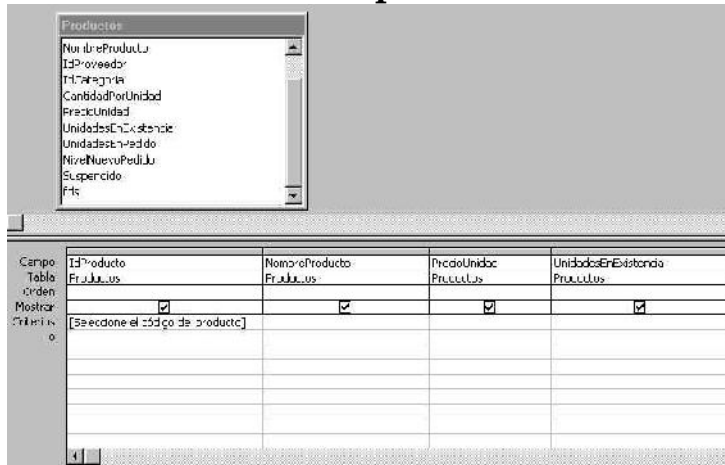
"No"
"Falso"
"Desactivado"
"0"

3.6 *Carácter comodín*

Como en MS-DOS o la mayoría de programas de Windows, podemos incluir los dos caracteres comodín (el asterisco * y el interrogante?) para crear una expresión. El interrogante puede sustituir un solo carácter en la posición que esté. El asterisco podrá sustituir a más de un carácter en la posición en que se encuentre. Por ejemplo:

- **M??** Buscará los datos que comiencen por M y luego tengan dos letras más
- **?a?a** buscará los datos cuya segunda y cuarta letra sea una a
- ***/12/97** buscará los datos de cualquier día del mes de diciembre del 97
- **An*** buscará los datos de lo que comience por la palabra An

4 Expresiones con Parámetros. Empleo de criterios de selección condicionados a la petición del usuario.



También en una consulta es posible condicionar la misma a los parámetros introducidos por el usuario. Por ejemplo en una consulta productos de la empresa se puede solicitar que se seleccione el código de aquellos productos para ver las existencias que hay en almacén.

Tal y como se muestra en la Ilustración anterior.

Otra forma de expresar las consultas en el campo criterio sería:

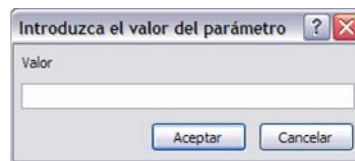
Como [Seleccione el código de producto] & '*' que nos permitirá invalidar selección nula.

Es decir, a menudo, en una consulta necesitamos utilizar **un valor que no es conocido en ese momento** sino que queremos que lo **introduzca el usuario cuando se ejecute la consulta**. Por ejemplo, queremos hacer una consulta para obtener los alumnos de una determinada Población, la población la introducirá el usuario cuando Access se lo pida.

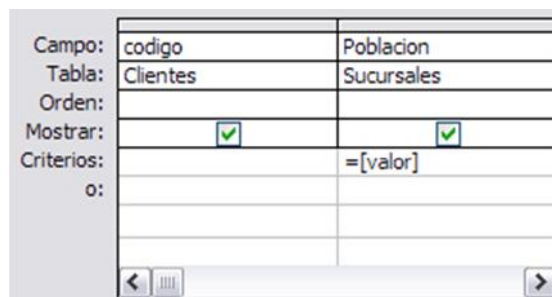
En este caso necesitamos **utilizar en nuestra consulta un parámetro**.

Un **parámetro** funciona de forma parecida a un campo de tabla, pero el valor que almacena lo introduce el usuario cuando se ejecuta la consulta.

En una consulta cuando utilizamos un nombre de campo que no está en el origen de datos, Access considera este campo como un parámetro y cuando se ejecuta la consulta nos pide **Introducir el valor del parámetro** mediante un cuadro de diálogo como este:

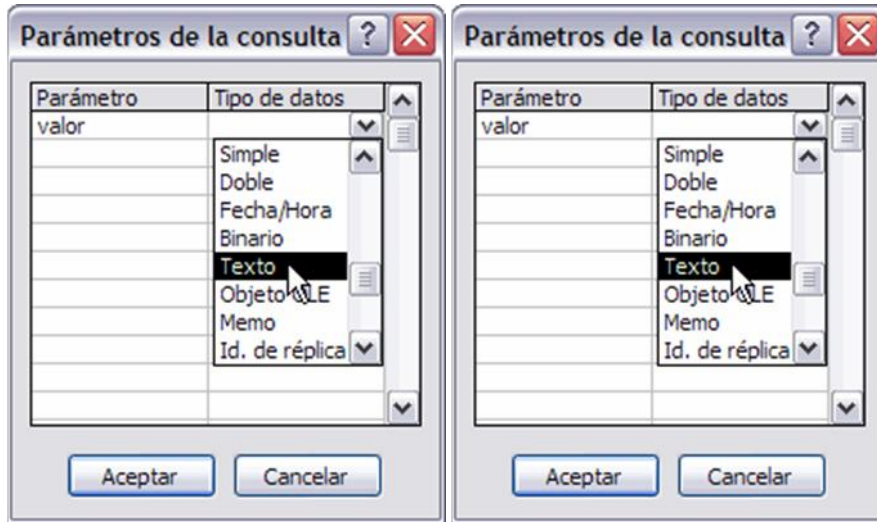


En el ejemplo anterior, en la consulta tendríamos que añadir una condición de búsqueda que especifique que la **Población** es igual al **Valor** a introducir, de esta manera



Ojo! cuando pongamos el nombre del parámetro **es importante escribirlo entre corchetes**, de lo contrario Access le añadirá comillas y no lo considerará como un nombre de parámetro sino como un valor.

Otra forma de utilizar un parámetro en una consulta es definiéndolo mediante la opción **Parámetros** del menú **Consulta**. En este caso, después de elegir la opción, se abre el cuadro de diálogo **Parámetros de la consulta** donde podemos indicar el **nombre del parámetro** y el **tipo de dato**.



5 Bibliografía

http://www.aulaclie.es/access2000/f_acces2000.htm
<http://www.aulafacil.com/Access/CursoAccess/CursoAccess.htm>
<http://www.aulaclie.es/access2003/>
<http://www.duiops.net/manuales/access/access.htm>
<http://www.programatium.com/access.htm>
<http://www.unav.es/cti/manuales/accessXP/indice.html>
http://www.aulaclie.es/access2003/t_7_6.htm
<http://www.alejandrox.com/2007/02/relaciones-en-access/>