

teleformacion.fer.es © ADR Infor SL  
ALBERTO CASTILLO ESTEBAS

# **SQL Server 2008. Índices. Vistas. © ADR Infor SL**

teleformacion.fer.es © ADR Infor SL  
ALBERTO CASTILLO ESTEBAS

teleformacion.fer.es © ADR Infor SL  
ALBERTO CASTILLO ESTEBAS

es © ADR Infor SL  
TILLO ESTEBAS

# Indice

<b>Nº- 9 SQL Server 2008. Índices y Vistas.....</b>	<b>1</b>
1 Índices.....	1
2 Mantenimiento de índices.....	1
3 Índices agrupados y no agrupados.....	2
3.1 Índices agrupados.....	2
3.2 Índices no agrupados.....	2
4 Creación de índices.....	3
5 Vistas.....	6
6 Creación de vistas.....	6
<b>Ejercicios.....</b>	<b>10</b>
Ejercicio 1.....	10
Ejercicio 2.....	10

# Nº- 9 SQL Server 2008. Índices y Vistas.

## 1 Índices.

Cuando quieres buscar el número de teléfono de una determinada persona en la guía de teléfonos, lo primero que hacemos es buscar por orden alfabético el número de página correspondiente. Lo mismo sucede cuando queremos buscar un determinado capítulo en un libro, primero acudimos al índice para encontrar rápidamente la página donde se encuentra.

Si no existiese un índice tendríamos que buscar el capítulo empezando por el principio del libro hasta dar con él. Las tablas en SQL Server funcionan de un modo muy parecido. Cuando creamos una tabla y comenzamos a insertar registros, estos se insertan según se van guardando, este es el único orden que podemos encontrar en una tabla. De este modo, cuando realizamos la búsqueda de un determinado registro, SQL Server tiene que ir leyendo cada registro hasta leerlos todos.

Como puedes imaginar, este tipo de búsqueda la cual se conoce como recorrido de tabla, es un proceso lento y costoso. Como se necesitará un acceso rápido de datos, debemos realizar una organización en las tablas que contienen esos datos.

Esta organización la realizaremos mediante la indización.

Los índices son estructuras que necesariamente tienen que ir unidas a una tabla o una vista. No podemos crear un índice, sin tener la tabla o vista al que vincularlo.

Por lo tanto en el momento de crear un índice debemos seleccionar la tabla o vista a la que pertenece, analizar la actividad en esa tabla o vista y cual es la finalidad para la que vamos a construir ese índice.

Un índice como hemos dicho, beneficia en gran medida la efectividad de las consultas, pero a su vez y de manera proporcional afecta en el rendimiento en las operaciones de inserción y actualización.

Los índices son estructuras creadas a partir de datos que ya existen en una columna o más de una tabla o vista, es decir no contiene ninguna información adicional, ya que es la misma que ya teníamos en la base de datos. Lo que hace al índice una herramienta eficaz, no es lo que almacena, sino el orden en el que se almacena esa información.

Este orden es implementado para acelerar las operaciones de búsqueda y selección, y con ello solucionamos el problema que hemos mencionado de almacenar la información en las tablas según se van insertando los registros. En el índice tendremos un campo o varios de la tabla pero ordenado correctamente para facilitar la búsqueda. Debe quedar claro que el índice no varía el orden de los registros en la tabla o vista, sino que se crea independientemente como otro objeto, como si fuese una segunda tabla auxiliar, la búsqueda se realiza sobre esta tabla ordenada y cuando encuentra el registro buscado nos devuelve la posición del mismo en la tabla o vista original, accediendo a él de un modo más rápido.

## 2 Mantenimiento de índices.

Del anterior capítulo llegamos a la conclusión de que un índice es un objeto creado a partir de la información de una o varias columnas de una tabla o vista, el índice almacena esos datos, pero ordenados eficazmente.

Pero en el momento de añadir un registro a nuestra tabla original, el índice que desactualizado y pierde su utilidad.

Al añadir o actualizar registros de las tablas o vistas en las que se han creado índices, éstos deben ser también actualizados. En un índice la información no se añade, entendiendo por añadir incluir un registro al final de la tabla. En un índice se insertan los registros en la posición correspondiente al orden fijado.

Como puedes imaginar, esta tarea consume recursos y memoria en el servidor, motivos suficientes para estudiar si el índice es rentable para una determinada tabla, o ganamos más si no lo creamos.

La principal norma que se suele seguir para decidir si crear un índice o no, es estimar el número de registros que tendremos en una tabla, si este número no es muy elevado es posible que el índice no sea necesario, en cambio si el número será elevado el índice nos será de gran ayuda. Por ejemplo es posible que en una empresa no muy grande, el número de empleados sea bajo y no vaya a incrementar demasiado en un futuro, pero en cambio el número de ventas que realiza la empresa si crecerá con frecuencia. Para la tabla empleados es posible que el índice no sea necesario, mientras que para la de ventas, será casi obligatorio.

### 3 Índices agrupados y no agrupados.

SQL Server tiene dos tipos de índices, tenemos dos tipos:

- Índices agrupados.
- Índices no agrupados.

Una tabla puede contar tener un índice agrupado, y uno o más índices no agrupados (o ninguno). Es decir, podemos tener en una misma tabla tantos índices como deseemos pero sólo uno de ellos debe ser agrupado.

#### 3.1 Índices agrupados.

Un índice agrupado ofrece las mismas ventajas que cualquier tipo de índice, pero además fija el orden de las filas en la tabla original a la cual pertenece.

Mientras no tengamos un índice agrupado los registros se van añadiendo uno seguido de otro, sin ningún tipo de orden. El índice facilita las búsquedas, pero las operaciones en ella necesitan buscar el lugar ordenado donde ubicar el nuevo registro a insertar, o incluso cambiar el orden en la misma.

Por norma general, el índice agrupado de una tabla siempre será el asociado a la clave principal, en caso de existir una. Si esta clave principal además es un número entero, mejora el rendimiento y el espacio ocupado en memoria. Si además este número entero es generado automáticamente activando la propiedad identity, los nuevos registros siempre se añadirán al final, y por lo tanto no será necesario buscar la posición donde insertar el nuevo registro, mejorando el rendimiento.

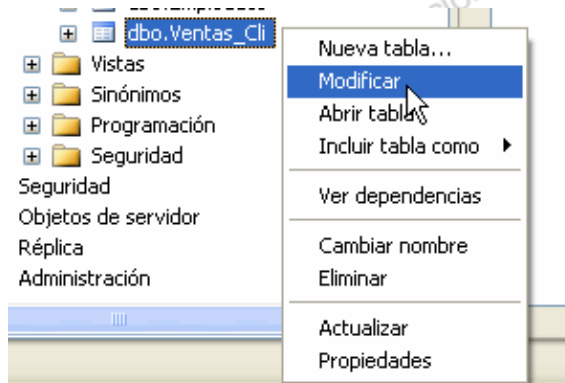
#### 3.2 Índices no agrupados.

Un índice no agrupado hace referencia a los registros de datos correspondientes mediante el índice agrupado.

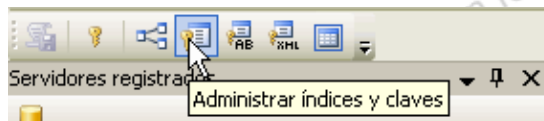
## 4 Creación de índices.

Para administrar los índices de una tabla usaremos el cuadro de diálogo "Índices y claves". El proceso es el siguiente:

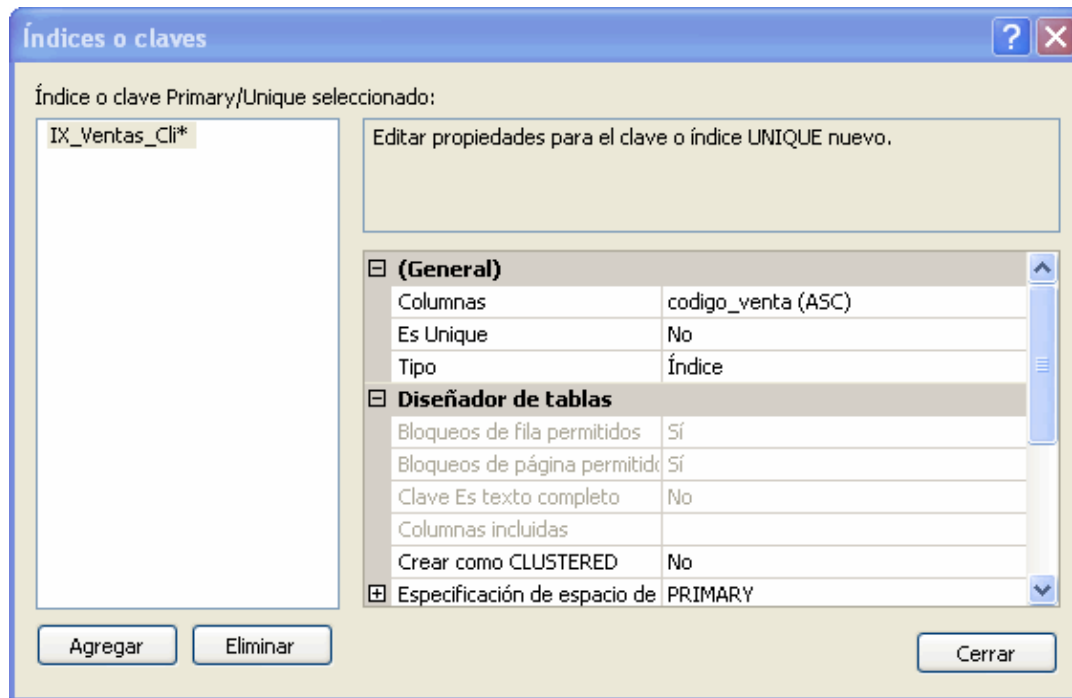
- Abrimos SQL Server Management Studio, abrimos la base de datos Ventas y pulsamos con el botón derecho sobre la tabla "Ventas\_Cli" seleccionando la opción "Modificar".



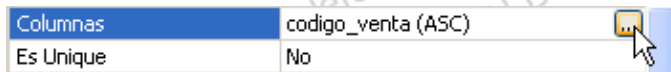
- Pulsamos sobre el botón "Administrar índices y claves" de la barra de herramientas de la parte superior.



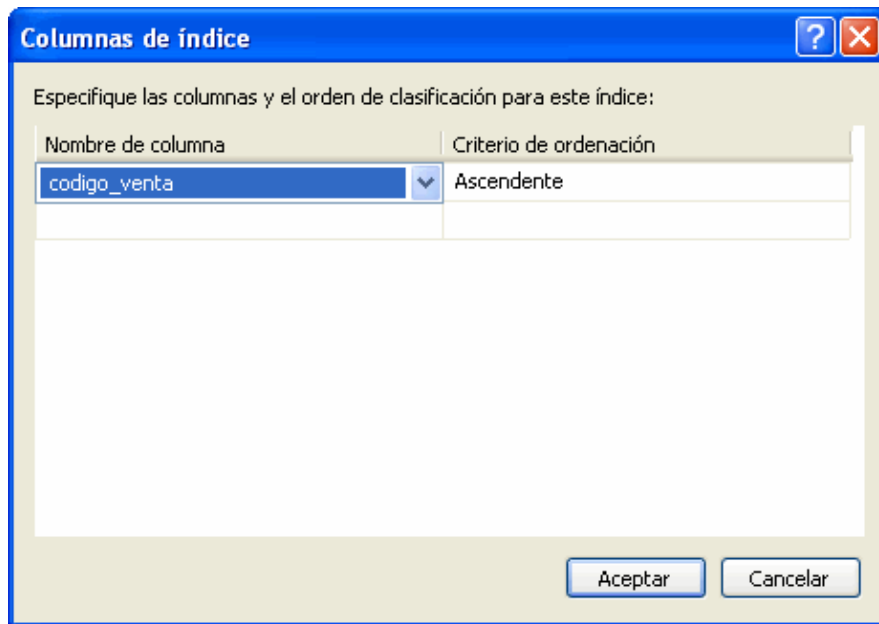
- Aparece el cuadro de diálogo "Índices y claves", desde esta ventana podemos hacer las principales tareas con el botón "Agregar" y "Eliminar", así como modificar las propiedades de los nuevos índices o los ya existentes.



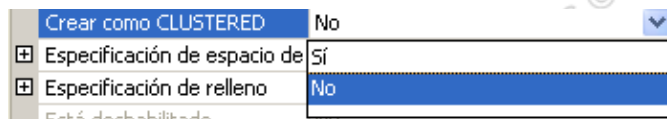
- Para crear un nuevo Índice pulsamos en el botón "Agregar", con lo que se añade a la izquierda un nuevo índice con un nombre por defecto, y a la derecha podremos cambiar las propiedades. Pulsamos sobre el botón derecho de la propiedad "Columnas" para mostrar el cuadro de diálogo donde se puede elegir las columnas que componen el índice.



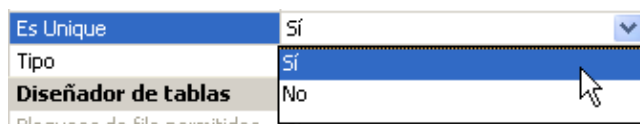
- Podemos elegir varias columnas, y establecer el orden de las mismas, mediante la propiedad "Criterio de ordenación".



- Entre otras propiedades de configuración tenemos la opción "Crear como CLUSTERED" (índice agrupado), en caso de existir ya un índice agrupado esta propiedad será inaccesible.



- Con la opción es UNIQUE, el índice con esta cláusula no habrá duplicaciones de valor.



- El resto de las opciones podemos dejarlas por defecto, controlan aspectos como son el relleno de nodos que forman el índice, recálculo de estadísticas de índices y grupos de archivos que forma el índice. La opción "Pasar por alto claves duplicadas" se activará si activamos la propiedad Es Unique, con el objetivo de ignorar la duplicación de claves.
- Al pulsar sobre el botón "Cerrar" se creará el índice y las operaciones sobre nuestra tabla serán mas eficaces. Podríamos crear un índice para otra columna por la que las búsquedas vayan a ser más frecuentes.



Creación de índices.

## 5 Vistas.

Las vistas las define Microsoft como una tabla virtual o como una consulta SELECT que se guarda.

Una vista representa los datos que están almacenados en una tabla. Algunas de las tablas que tenemos en la base de datos puede llegar a tener un tamaño importante, con miles y miles de registros. Lo más probable es que no necesitemos acceder a la totalidad de esos registros al mismo tiempo, podemos almacenar un grupo de esos registros con los que vayamos a trabajar más frecuentemente para agilizar las consultas en gran medida.

Otra ventaja que nos ofrecen las vistas es la posibilidad de crear una vista que muestre datos de varias tablas. Más adelante esta vista puede ser utilizada como si una tabla se tratase.

## 6 Creación de vistas.

A continuación explicaremos el proceso de creación de una vista que nos muestre los siguientes datos de la base de datos Northwind:

Columna	Tabla	Descripción
CompanyName	Customers	Nombre de la compañía
ContactName	Customers	Nombre representante.
OrderDate	Orders	Fecha de pedido.
FirstName	Employees	Nombre del empleado.
LastName	Employees	Apellidos del empleado.

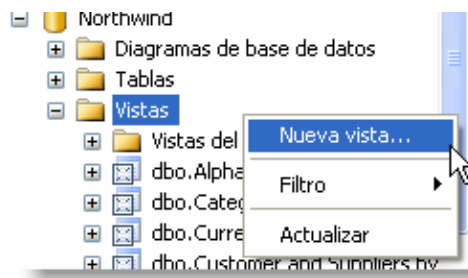
Estos datos se mostrarán con las siguientes características:

- Mostraremos estos valores para las ventas producidas en el año 1997.
- Los registros se mostrarán ordenados ascendentemente de menor a mayor por el nombre de cliente.

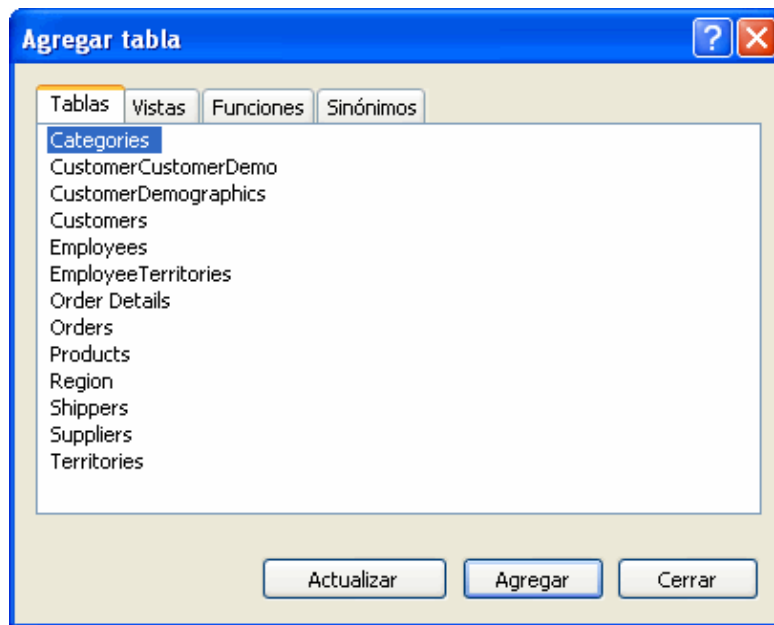
El proceso será el siguiente:

- Desplegamos la base de datos "Northwind" y pulsamos con el botón derecho sobre la carpeta "Vistas" y seleccionamos la opción "Nueva Vista".

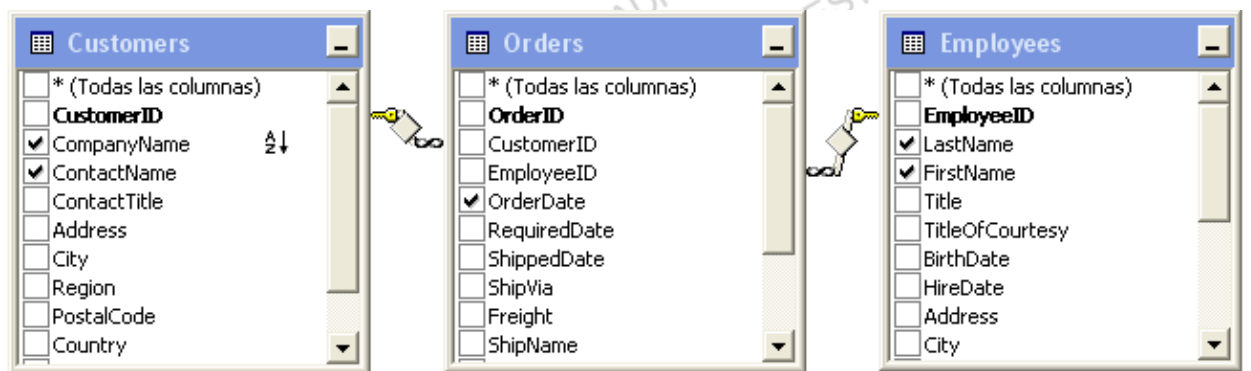




- Al pulsar esta opción SQL Server nos guiará por un asistente para crear la nueva vista. El primer paso es seleccionar las tablas que incluiremos en la consulta que compone la vista. Seleccionamos cada tabla y la vamos agregando.



- En la siguiente ventana que nos muestra una vez cerrada la anterior ventana, tenemos a nuestra disposición las tablas incluso con las relaciones existentes entre ellas. Utilizaremos estas tablas para seleccionar las columnas que queremos mostrar en la vista. Puedes observar como la sentencia se va construyendo automáticamente.



## SQL Server 2008. Índices. Vistas.

- En el segundo panel, encontramos el modo de modificar las características de la consulta mediante sus columnas, pudiendo aplicar diferentes filtros que se añadirán en la cláusula WHERE de nuestra consulta., la preferencia de orden de esa columna (Criterio de ordenación), el tipo de orden, si es un campo que debe mostrarse y el alias si deseamos renombrar la columna. En la siguiente figura, se muestran la configuración de estas características para la consulta que pretendemos.

Columna	Alias	Tabla	Resu...	Tipo de orden	Criterio de ordenación	Filtro
CompanyName		Customers	<input checked="" type="checkbox"/>	Ascendente	1	
ContactName		Customers	<input checked="" type="checkbox"/>			
OrderDate		Orders	<input checked="" type="checkbox"/>			>= '01/01/1997'
FirstName		Employees	<input checked="" type="checkbox"/>			
LastName		Employees	<input checked="" type="checkbox"/>			
OrderDate		Orders	<input type="checkbox"/>			<= '31/12/1997'

- El código que hemos ido construyendo y se ha generado automáticamente es el siguiente:

```
SELECT TOP (100) PERCENT dbo.Customers.CompanyName, dbo.Customers.ContactName, dbo.Orders.OrderDate, dbo.Employees.FirstName,
    dbo.Employees.LastName
FROM    dbo.Customers INNER JOIN
        dbo.Orders ON dbo.Customers.CustomerID = dbo.Orders.CustomerID INNER JOIN
        dbo.Employees ON dbo.Orders.EmployeeID = dbo.Employees.EmployeeID
WHERE   (dbo.Orders.OrderDate >= CONVERT(DATETIME, '1997-01-01 00:00:00', 102)) AND (dbo.Orders.OrderDate <= CONVERT(DATETIME,
    '1997-12-31 00:00:00', 102))
ORDER BY dbo.Customers.CompanyName
```

Este diseñador de consultas puede ser muy eficaz para crear sentencias, pero no siempre cumple con nuestro objetivo y en frecuentes ocasiones seremos nosotros mismos los que tengamos que construir estas consultas por eso es tan importante tener unos conocimientos en lenguaje SQL. Nunca el diseñador de consultas de SQL Server puede abarcar las necesidades que cubriremos con nuestros conocimientos.

- En esta misma ventana podemos ejecutar la consulta y comprobar que los resultados son los deseados

	CompanyName	ContactName	OrderDate	LastName	FirstName
▶	Alfreds Futterkiste	Maria Anders	25/08/1997 0:0...	Suyama	Michael
	Alfreds Futterkiste	Maria Anders	03/10/1997 0:0...	Peacock	Margaret
	Alfreds Futterkiste	Maria Anders	13/10/1997 0:0...	Peacock	Margaret
	Alfreds Futterkiste	Maria Anders	15/01/1998 0:0...	Davolio	Nancy

- Pulsamos sobre el botón "Guardar" de la base de datos y le damos como nombre a la vista "ventas1997"
- Por último para probar la vista que hemos diseñado, abrimos una nueva consulta sobre la base de datos "Northwind" y realizamos una consulta SELECT, ya que como hemos dicho una vista puede ser tratada como una tabla:

SELECT \* FROM ventas1997

	CompanyName	ContactName	OrderDate
1	Eastern Connection	Ann Devon	1997-01-01 00:00:00.000
2	Rattlesnake Canyon Grocery	Paula Wilson	1997-01-01 00:00:00.000
3	Ernst Handel	Roland Mendel	1997-01-02 00:00:00.000
4	Ernst Handel	Roland Mendel	1997-01-03 00:00:00.000
5	Magazzini Alimentari Riuniti	Giovanni Rovelli	1997-01-03 00:00:00.000
6	LIND-Delicatesses	Felipe Izquierdo	1997-01-06 00:00:00.000
7	Queen Cozinha	Lúcia Carvalho	1997-01-07 00:00:00.000
8	Otilies Käseladen	Henriette Pfalzheim	1997-01-07 00:00:00.000
9	Folies gourmandes	Martine Rancé	1997-01-08 00:00:00.000
10	Océano Atlántico Ltda.	Yvonne Moncada	1997-01-09 00:00:00.000
11	Bottom-Dollar Markets	Elizabeth Lincoln	1997-01-10 00:00:00.000
12	Bottom-Dollar Markets	Elizabeth Lincoln	1997-01-10 00:00:00.000
13	Wartian Herkku	Pirkko Koskitalo	1997-01-13 00:00:00.000
14	La maison d'Asie	Annette Roulet	1997-01-14 00:00:00.000
15	Familia Arquibaldo	Aria Cruz	1997-01-14 00:00:00.000

Si observamos el resultado obtenido al realizar la consulta en la vista, la consulta sobre la misma no ha respetado el orden de los resultados, de todos modos he creído conveniente aplicarlo en la vista para ver como ordenar resultados utilizando el diseñador de consultas.



# Ejercicios

## Ejercicio 1

Supongamos que las consultas que se realizan en la tabla de usuarios por el campo apellidos, son los suficientemente frecuentes como para crear un índice de orden ascendente en ese campo.

## Ejercicio 2

Vamos a crear una vista para facilitar las consultas de tal modo que obtengamos la información detallada de las fichas que tengan como nivel de actividad 2.

Esta consulta devolverá todos los datos que tenemos alojados en nuestras tres tablas, a excepción de los códigos.