

Administración SQL

Básica

Segunda parte

Índices

SQL Server accede a los datos de dos maneras:

- Recorriendo las tablas; comenzando el principio y extrayendo los registros que cumplen las condiciones de la consulta.
- Empleando índices; recorriendo la estructura de árbol del índice para localizar los registros y extrayendo los que cumplen las condiciones de la consulta.

Los índices se emplean para facilitar la obtención de información de una tabla.

El índice de una tabla desempeña la misma función que el índice de un libro: permite encontrar datos rápidamente.

El objetivo de un índice es **acelerar la recuperación de información**.

La indexación es una técnica que optimiza el acceso a los datos, mejora el rendimiento acelerando las consultas y otras operaciones.

Es útil cuando la tabla contiene miles de registros, **cuando se realizan operaciones de búsqueda, ordenamiento y agrupamiento y cuando se combinan varias tablas**.

La desventaja es que **consume espacio en el disco** en disco y genera costo de mantenimiento (tiempo y recursos).

Es importante identificar el o los campos por los que sería útil crear un índice, aquellos campos por los cuales se **realizan búsqueda con frecuencia**: claves primarias, claves externas o campos que combinan tablas.

SQL Server permite crear dos tipos de índices: 1) agrupados (clustered) y 2) no agrupados (nonclustered).

- Un **ÍNDICE AGRUPADO** (CLUSTERED) determina la secuencia de almacenamiento de los registros en una tabla. Se almacenarán físicamente en el orden especificado por el índice. Se utilizan para campos por los que se realizan búsquedas con frecuencia o se accede siguiendo un orden. Una tabla sólo puede tener UN índice agrupado. Suele ser la Primary key.
- Un **ÍNDICE NO AGRUPADO** (NONCLUSTERED) genera una estructura aparte con la ordenación de los registros, los datos se almacenan en un lugar diferente al del índice, los punteros indican el lugar de almacenamiento de los elementos indizados en la tabla. Es eficiente aunque más lento que el agrupado. Si no se especifica un tipo de índice, de modo predeterminado será no agrupado.

Los campos de tipo text, ntext e image no se pueden indizar.

Crear un índice:

```
create TIPODEINDICE index NOMBREINDICE  
on TABLA(CAMPO);
```

Con TipoDeIndice:

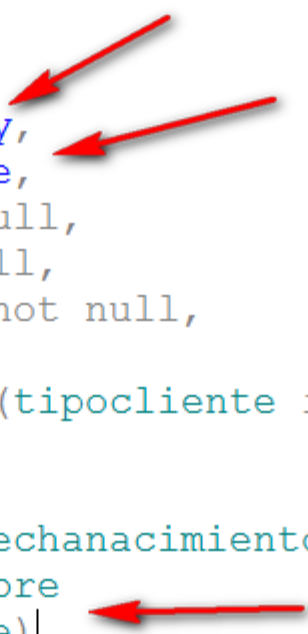
CLUSTERED

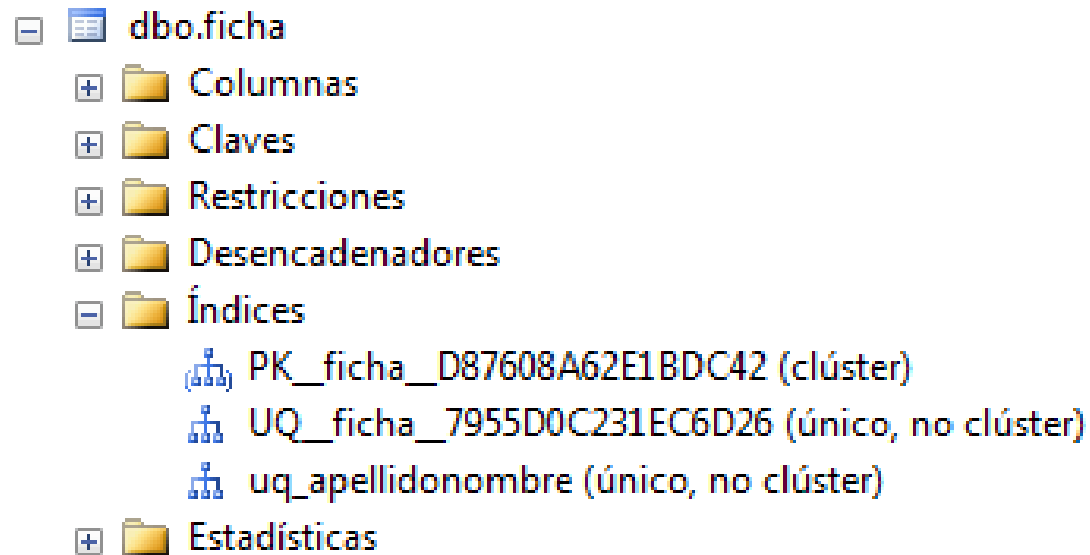
NONCLUSTERED

Para esta tabla generará automáticamente unos índices...

```
= if object_id('ficha') is not null
  drop table ficha;

= create table ficha(
  dni varchar(8) primary key,
  codigosocio integer unique,
  apellido varchar(50) not null,
  nombre varchar(50) not null,
  fechanacimiento datetime not null,
  fechaingreso datetime,
  tipocliente char(1) check(tipocliente in ('a','b','c')),
  Direccion varchar(100),
  constraint ck_fechas
    check(fechaingreso>=fechanacimiento),
  constraint uq_apellidonombre
    unique(apellido,nombre)|
)
go
```





Para la clave principal clúster

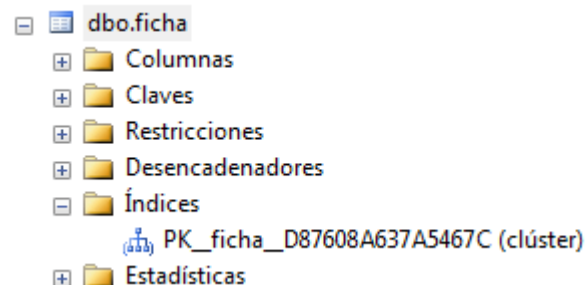
Para cada unique un no clúster

Creamos la tabla sólo con la clave principal...

```
if object_id('ficha') is not null
    drop table ficha;

create table ficha(
dni varchar(8) primary key,
codigosocio integer,
apellido varchar(50) not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1) check(tipocliente in ('a','b','c')),
Direccion varchar(100),
constraint ck_fechas
    check(fechaingreso>=fechanacimiento)
)
go
```

Genera sólo un índice clúster



Si intentamos crear un índice clúster nos indicará que no se puede porque ya hay uno.

```
create clustered index ix_codigosocio  
on ficha(codigosocio);  
go
```



Mensajes

Mens. 1902, Nivel 16, Estado 3, Línea 1
No se pueden crear varios índices clúster en tabla 'ficha'.

Podemos crearlo
no clúster.

```
create nonclustered index ix_codigosocio  
on ficha(codigosocio);  
go
```



Mensajes

Comandos completados correctamente.



Índices

ix_codigosocio (no único, no clúster)
PK_ficha_D87608A637A5467C (clúster)

Para ver los índices de una tabla haremos...

sp_helpindex tabla

Para eliminar un índice haremos...

drop index tabla.nombreindice

Vemos los índices de la tabla.

```
exec sp_helpindex ficha;  
go
```

	index_name	index_description	index_keys
1	ix_codigosocio	nonclustered located on PRIMARY	codigosocio
2	PK_ficha__D87608A637A5467C	clustered, unique, primary key located on PRIMARY	dni

Para eliminarlo...

```
drop index ficha.ix_codigosocio;  
go
```

Podemos hacer que una clave primaria genere un índice no clúster.

En la creación de la tabla indicaríamos...


campo tipo primary key nonclustered


o bien

primary key nonclustered (lista de campos)

```
if object_id('ficha') is not null
    drop table ficha;

create table ficha(
dni varchar(8) primary key nonclustered,
codigosocio integer,
apellido varchar(50) not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1) check(tipocliente in ('a','b','c')),
Direccion varchar(100),
constraint ck_fechas
    check(fechaingreso>=fechanacimiento)
)
```



[-]  Índices

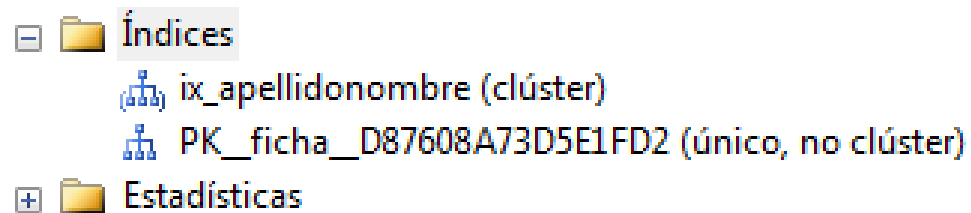
 PK_ficha_D87608A73D5E1FD2 (único, no clúster)

[+]  Estadísticas



En este caso podríamos crear un índice clúster para otro campo.

```
create clustered index ix_apellidonombre  
on ficha(apellido,nombre);  
go  
|
```



Podemos regenerar el índice para mejorar su velocidad o cuando detectemos errores.

Se hace con la misma sentencia añadiendo al final `with drop_existing`:

```
create TIPODEINDICE index NOMBREINDICE  
on TABLA(CAMPO)  
with drop_existing;
```


```
create clustered index ix_apellidonombre  
on ficha(apellido,nombre)  
with drop_existing;  
go
```


A la creación del índice podemos añadirle el atributo unique para que no permita valores duplicados.

```
create unique clustered index ix_apellidonombre  
on ficha(apellido,nombre)  
with drop_existing;  
go
```

Si intentamos añadir un registro con el par apellido/nombre duplicado no lo permitirá...

```
insert into ficha
  values ('4444444W',123,'Juan','Pérez','10/01/1969','01/01/2004',
        'a','C/primera nº 6');
insert into ficha
  values ('44444446Z',125,'Juan','Pérez','10/01/1969','01/01/1980',
        'a','C/primera nº 6');
```

 Mensajes

(1 filas afectadas)

Mens. 2601, Nivel 14, Estado 1, Línea 4

No se puede insertar una fila de clave duplicada en el objeto 'dbo.ficha' con el índice único 'ix_apellidonombre'.

Se terminó la instrucción.

Crear Bases de Datos

Ver características de la Base de datos:

```
exec sp_helpdb fichaspersonales  
go
```

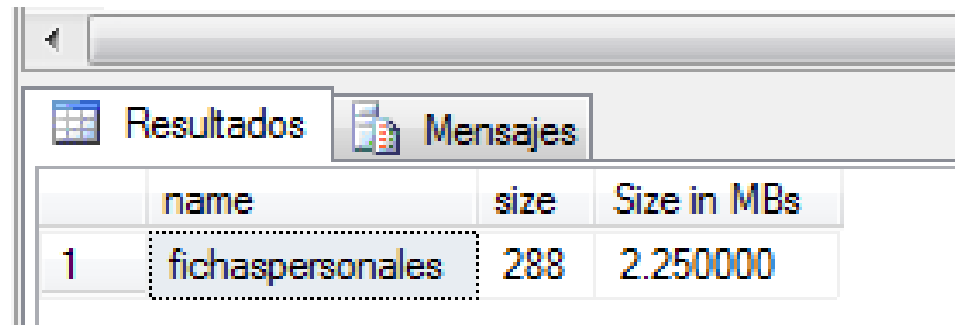
Resultados Mensajes

	name	db_size	owner	dbid	created	status	compatibility_level
1	fichaspersonales	2.81 MB	arodpes-PC\arodpes	118	Mar 30 2014	Status=ONLINE, Updateability=READ_WRITE, UserAccess=MULTI_USER, Recovery=SI...	100

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	fichaspersonales	1	c:\Program Files\Microsoft SQL Server\MSSQL10_50...	PRIMARY	2304 KB	Unlimited	1024 KB	data only
2	fichaspersonales_log	2	c:\Program Files\Microsoft SQL Server\MSSQL10_50...	NULL	576 KB	2147483648 KB	10%	log only

Ver tamaño de la Base de Datos:

```
SELECT name, size, size*1.0/128 AS [Size in MBs]  
FROM sys.master_files  
WHERE name = N'fichaspersonales';
```



The screenshot shows a SQL Server Enterprise Manager window. At the top, there is a toolbar with a 'Results' icon (a grid) and a 'Messages' icon (an envelope). Below the toolbar, there are two tabs: 'Resultados' (Results) and 'Mensajes' (Messages). The 'Resultados' tab is active, displaying a table with the following data:

	name	size	Size in MBs
1	fichaspersonales	288	2.250000

Para crear la base de datos pudiendo tocar sus características:

```
CREATE DATABASE AdmBD2
ON
( NAME = AdmBD2_dat,
  FILENAME = 'C:\SQL\AdmBD2.mdf',
  SIZE = 10MB,
  MAXSIZE = 50MB,
  FILEGROWTH = 5MB )
LOG ON
( NAME = AdmBD2_log,
  FILENAME = 'C:\SQL\AdmBD2_log.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB ) ;
```

Tamaño puede ir en [KB | MB | GB | TB]

Tenemos dos archivos:

- El archivo de datos, donde se almacena toda la información.
- El archivo con el Log de transacciones, donde se almacenan las modificaciones que se realizan en la Base de Datos.

Las características que definimos para cada archivo:

SIZE = tamaño inicial

MAXSIZE = tamaño máximo

FILEGROWTH = incremento de tamaño cada vez que se llene el espacio ocupado

Hay que hacer una previsión del tamaño de la Base de datos, para crearla con el tamaño más aproximado, con lo que conseguiremos que el espacio en disco duro esté compacto y así mejoraremos el rendimiento.

Ejercicio

Crear Base de datos AdmBD3 guardando el archivo de datos y el de transacciones en C:\DatosSQL, con un tamaño inicial de 20 MegaBytes, incremento de 5 en 5 MegaBytes y tamaño máximo de 50 MegaBytes.

Modificar la estructura de una tabla: alter
table

Alter table permite modificar la estructura de una tabla.

Podemos utilizarla para agregar, modificar y eliminar campos de una tabla, sin afectar al resto de sus datos.

También lo podemos usar con las constraints, como ya hemos visto.

Para los siguientes ejercicios partiremos de lo siguiente:

```
use fichaspersonales
go

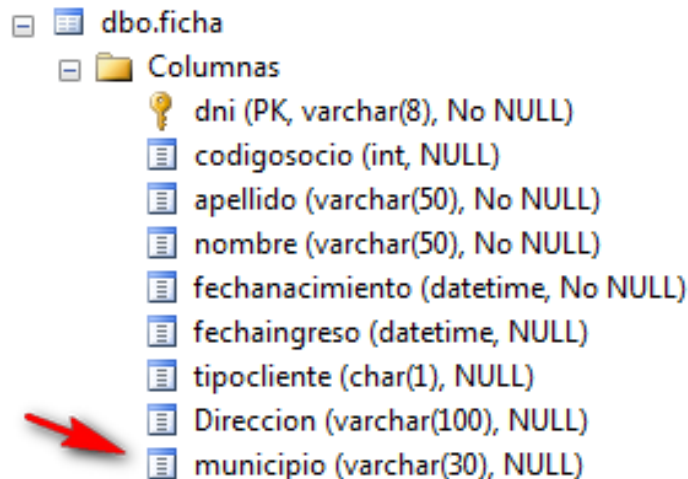
if object_id('ficha') is not null
    drop table ficha;
go

create table ficha(
dni varchar(8) primary key,
codigosocio integer unique,
apellido varchar(50) not null,
nombre varchar(50) not null,
fechanacimiento datetime not null,
fechaingreso datetime,
tipocliente char(1) check(tipocliente in ('a','b','c')),
Direccion varchar(100),
constraint ck_fechas
    check(fechaingreso>=fechanacimiento)
);
go
```

Añadir un campo

```
alter table NOMBRETABLA  
add NOMBRENUEVOCAMPO DEFINICION;
```

```
alter table ficha  
add municipio varchar(30);  
go
```



Para ver la estructura también se puede usar sp_columns.

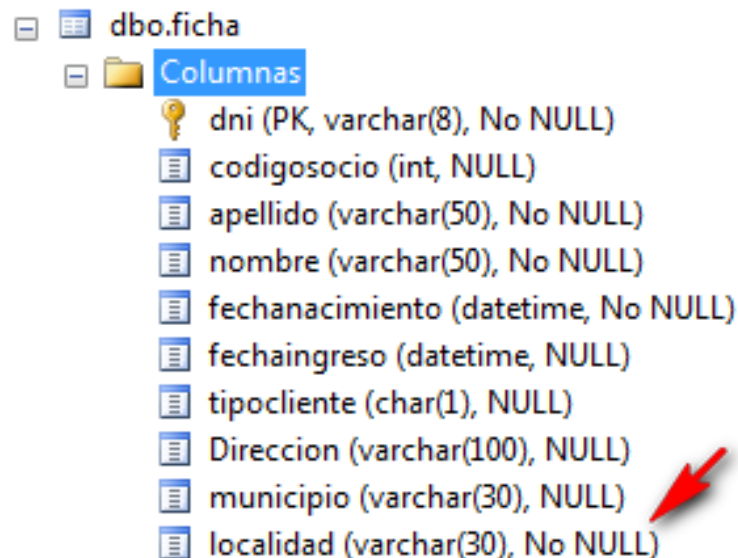
```
exec sp_columns tabla;  
go
```

```
exec sp_columns ficha;  
go
```

	TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	PRECISION	LENGTH	SCALE	RADIX	NULLA
1	fichaspersonales	dbo	ficha	dni	12	varchar	8	8	NULL	NULL	0
2	fichaspersonales	dbo	ficha	codigosocio	4	int	10	4	0	10	1
3	fichaspersonales	dbo	ficha	apellido	12	varchar	50	50	NULL	NULL	0
4	fichaspersonales	dbo	ficha	nombre	12	varchar	50	50	NULL	NULL	0
5	fichaspersonales	dbo	ficha	fechanacimiento	11	datetime	23	16	3	NULL	0
6	fichaspersonales	dbo	ficha	fechaingreso	11	datetime	23	16	3	NULL	1
7	fichaspersonales	dbo	ficha	tipocliente	1	char	1	1	NULL	NULL	1
8	fichaspersonales	dbo	ficha	Direccion	12	varchar	100	100	NULL	NULL	1
9	fichaspersonales	dbo	ficha	municipio	12	varchar	30	30	NULL	NULL	1

Si el campo debe ser NOT NULL tendremos que definirle valor default o primero crearlo sin el NOT NULL, cargar los datos en la tabla y después modificar el campo colocando el atributo NOT NULL.

```
alter table ficha  
add localidad varchar(30) not null default 'Desconocida';  
go
```



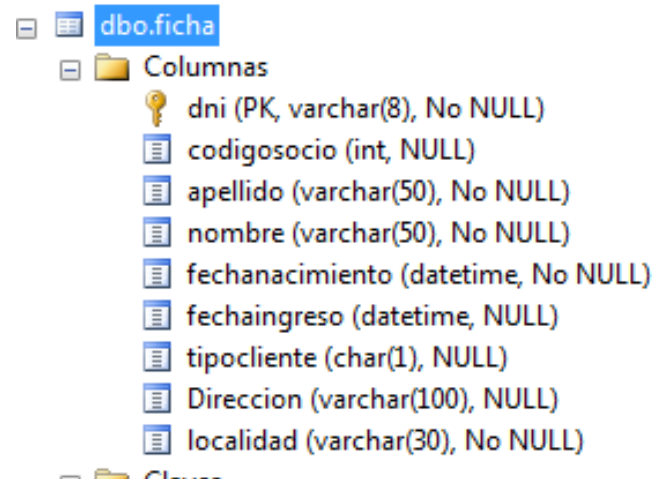
Para **eliminar un campo** usamos drop column:

```
alter table tabla  
    drop column campo;
```

```
alter table ficha  
drop column municipio;  
go
```

Ya no aparece el campo municipio. Tampoco podremos recuperar la información que almacenáramos en él.

No pide confirmación!!!!



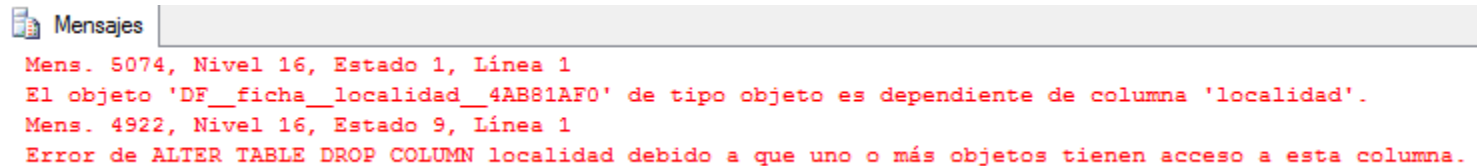
También eliminar varios campos:

```
alter table tabla  
    drop column campo1,campo2;
```

No pueden eliminarse los campos que son usados por un índice o tengan restricciones.

No puede eliminarse un campo si es el único en la tabla.

```
alter table ficha
drop column localidad;
go
```

The screenshot shows a 'Mensajes' (Messages) window in SQL Server. It contains two messages. The first message, at Mens. 5074, Nivel 16, Estado 1, Línea 1, states: 'El objeto 'DF__ficha__localidad__4AB81AF0' de tipo objeto es dependiente de columna 'localidad'.'. The second message, at Mens. 4922, Nivel 16, Estado 9, Línea 1, states: 'Error de ALTER TABLE DROP COLUMN localidad debido a que uno o más objetos tienen acceso a esta columna.'.

Mensajes

Mens. 5074, Nivel 16, Estado 1, Línea 1
El objeto 'DF__ficha__localidad__4AB81AF0' de tipo objeto es dependiente de columna 'localidad'.

Mens. 4922, Nivel 16, Estado 9, Línea 1
Error de ALTER TABLE DROP COLUMN localidad debido a que uno o más objetos tienen acceso a esta columna.

Hay una constraint (el valor default) asociada al valor.

Hay que eliminarla primero sacando el nombre del error.

```
alter table ficha
drop constraint DF__ficha__localidad__4AB81AF0;
alter table ficha
drop column localidad;
go
```


Modificar campos

```
alter table NOMBRETABLA  
alter column CAMPO NUEVADEFINICION;
```

```
]alter table ficha  
-alter column apellido varchar(100) null;  
-go
```

Aumentando el tamaño y permitiendo valores NULL

No se permiten modificar:

- campos de tipo text, image, ntext y timestamp.
- un campo que es usado en un campo calculado.
- campos que son parte de índices o tienen restricciones, a menos que el cambio no afecte al índice o a la restricción, por ejemplo, se puede ampliar la longitud de un campo de tipo carácter.
- agregando o quitando el atributo "identity".
- campos que afecten a los datos existentes cuando una tabla contiene registros (ejemplo: un campo contiene valores nulos y se pretende redefinirlo como "not null"; un campo int guarda un valor 300 y se pretende modificarlo a tinyint, etc.).

Campos calculados

Un campo calculado es un campo que no se almacena físicamente en la tabla. SQL Server emplea una fórmula que detalla el usuario al definir dicho campo para calcular el valor según otros campos de la misma tabla.

En la creación de la tabla se añadiría:

campo as expresión

Igualmente en el alter table

alter table tabla

add campo as expresión

```
|if object_id('ficha') is not null
| drop table ficha;
|go

|create table ficha(
| dni varchar(8) primary key,
| codigosocio integer unique,
| apellido varchar(50) not null,
| nombre varchar(50) not null,
| fechanacimiento datetime not null,
| fechaingreso datetime,
| tipocliente char(1) check(tipocliente in ('a', 'b', 'c')),
| Direccion varchar(100),
| mesnacimiento as datename(month, fechanacimiento),
| primeraletranombre as substring(nombre, 1, 1),
| constraint ck_fechas
|     check(fechaingreso >= fechanacimiento)
|);
```

Los campos calculados no se almacenan, se calculan cada vez que accedamos al registro. Están disponibles en el select.

Con el alter table...

```
|  
alter table ficha  
    add letradni as left(dni,1);  
go
```

