**OPCIONES DE INTERCALACIÓN**: Pueden establecerse a nivel de servidor, de base de datos, de columna o de expresión:

(https://learn.microsoft.com/es-es/sql/relational-databases/collations/collation-and-unicode-support?view=sql-server-ver16#SQL-collations)

|  |  |
| --- | --- |
| BIN | Orden binario. |
| BIN2 | Ordenación punto de código binario. |
| CI | No distingue entre mayúsculas y minúsculas. |
| CS | Sí distingue entre mayúsculas y minúsculas. |
| AI | No distingue acentos. |
| AS | Sí distingue acentos. |
| KS | Sí distingue Kana. |
| WS | Sí distingue ancho. |
|  | En el caso de KS y WS, su ausencia indica que no distingue Kana ni ancho respectivamente. |

**DIFERENTES TIPOS DE DATOS EN SQL SERVER**

**Alfanuméricos:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | TAMAÑO FIJO. SE DEBE UTILIZAR CUANDO SEPAMOS QUE LOS DATOS QUE ALBERGARÁ SERÁN SIEMPRE DE LA MISMA LONGITUD | TAMAÑO VARIABLE.  SE DEBE UTILIZAR CUANDO DESCONOCEMOS LA LONGITUD DE LOS DATOS QUE SE ALMACENARÁN | ARGUMENTO DEFINE EL TAMAÑO DE LA CADENA EN BYTES | ARGUMENTO DEFINE EL TAMAÑO DE LA CADENA EN PARES DE BYTES | MÁXIMO VALOR DEL ARGUMENTO |  |  |
| CHAR | ✔ |  | ✔ |  | 8000 |  |  |
| VARCHAR |  | ✔ | ✔ |  | 8000 - MAX |  |  |
| NCHAR | ✔ |  |  | ✔ | 4000 |  |  |
| NVARCHAR |  | ✔ |  | ✔ | 4000 - MAX |  |  |
| TEXT | SERÁ ELIMINADO Y SE RECOMIENDA SUSTITUIR POR VARCHAR(MAX) | | | | | | |
| NTEXT | SERÁ ELIMINADO Y SE RECOMIENDA SUSTITUIR POR NVARCHAR(MAX) | | | | | | |

CHAR o NCHAR se deben utilizar en campos cuyo contenido sepamos que será siempre de la misma longitud o con variaciones mínimas, por ejemplo, la matrícula de un coche.

Por el contrario, VARCHAR o NVARCHAR los utilizaremos en campos cuya longitud puede ser muy diferente de unos registros a otros, por ejemplo, una dirección postal.

NCHAR y NVARCHAR sólo deberían utilizarse en aplicaciones multilenguaje en las que se vayan a utilizar caracteres Unicode.

**Fechas:**

Date: Sólo almacena la fecha, es decir: año, mes y día

Time(n): Sólo almacena la hora, es decir: horas, minutos, segundos y hasta 7 dígitos de nanosegundos. Donde n es el número de dígitos que se usarán.

SmallDateTime: almacena fecha y hora sin nanosegundos. Además, los segundos siempre son cero. Por lo tanto, la precisión es de un minuto.

DateTime: Almacena fecha y hora con milisegundos. Rango de fechas: 01/01/1753 a 31/12/9999. Rango de horas: 00:00:00 a 23:59:59.997

DateTime2(n): Almacena fecha y hora con milisegundos. Rango de fechas: 01/01/0001 a 31/12/9999. Rango de horas: 00:00:00 a 23:59:59.9999999. n es el número de dígitos que se usarán.

DateTimeOffset(n): es exactamente igual que DateTime2 pero añade la zona horaria.

En resumen, considero que estamos utilizando el tipo DateTime siempre por defecto sin tener en cuenta los demás y sin tener en cuenta el tipo de dato que se desea almacenar. Muchos de los campos que definimos como DateTime deberían ser Date y ahorraríamos 5 bytes en cada uno de ellos. Sólo con los campos fecha\_creacion, hora\_creacion, fecha\_actua y hora\_actua podríamos ahorrar como mínimo 16 bytes por fila.

IMAGE: SERÁ ELIMINADO Y SE RECOMIENDA SUSTITUIR POR VARBINARY(MAX)