

## Tipi di dati in MS SQL Server

(secondo lo standard TRANSACT-SQL)

### Dati di tipo Numerico

Tra questi tipi di dato rientrano la definizione di numeri sui quali è possibile effettuare operazioni matematiche.

<b>TINYINT</b>	Rappresenta un <b>valore numerico non negativo compreso tra 0 e 255</b> ed occupa la dimensione pari a <b>1 byte</b> , viene spesso usato anche per archiviare una Enumerazione.
<b>SMALLINT</b>	Rappresenta un <b>valore numerico compreso tra -32768 e 32767</b> . E' il classico intero semplice ed occupa la dimensione di <b>2 byte</b> .
<b>INT</b>	Rappresenta un <b>valore numerico compreso tra -2.147.483.648 e 2.147.483.647</b> e occupa la dimensione di <b>4 byte</b> .
<b>BIGINT</b>	Rappresenta un <b>valore numerico compreso tra <math>-(2^{64})</math> e <math>+(2^{64})-1</math></b> ossia compreso tra -9,223,372,036,854,775,808 e +9,223,372,036,854,775,807. Occupa la dimensione di <b>8 byte</b> .
<b>DECIMAL(p,s)</b> <b>NUMERIC(p,s)</b>	Rappresenta un <b>numero decimale in virgola fissa</b> . Il parametro " <b>p</b> " indica la <b>precisione</b> (impostabile fra 1 e 38), cioè il numero totale di cifre, mentre " <b>s</b> " indica le <b>cifre dopo la virgola</b> . Occupa una dimensione che può variare <b>da 5 a 17 byte</b> , a seconda del valore di "p".
<b>REAL</b>	Rappresenta un <b>numero reale in virgola mobile</b> ed occupa una dimensione di <b>4 byte</b>
<b>FLOAT(p)</b>	Rappresenta un <b>numero reale in virgola mobile</b> , permettendo a differenza di REAL di specificarne la precisione tramite il parametro "p". Per $p \leq 25$ abbiamo precisione singola ed occupa <b>4 byte</b> , per p maggiore di 25 la precisione invece è doppia occupando <b>8 byte</b> .
<b>MONEY</b>	Rappresenta un <b>dato numerico indicante un valore monetario</b> . Sostanzialmente equivale a un tipo DECIMAL di <b>8 byte</b> con un arrotondamento a <b>4 cifre dopo la virgola</b> . Offre quindi valori compresi fra -922.337.203.685.477,5808 a 922.337.203.685.477,5807
<b>SMALLMONEY</b>	Rappresenta un <b>dato numerico indicante un valore monetario</b> equivalente al tipo MONEY ma con una occupazione <b>4 byte</b> e valori ammissibili compresi fra -214.748,3648 a 214.748,3647

### Dati di tipo Testuale

Tra questi tipi di dato rientrano le stringhe, ossia quei dati che permettono di definire un testo, anche particolarmente lungo o testi con lunghezze molto variabili, oltre alle stringhe binarie per la memorizzazione di dati multimediali.

<b>CHAR(n)</b>	Rappresenta una <b>stringa testuale fissa di caratteri</b> , dove "n" ne indica la lunghezza (max 8000). La dimensione occupata è pari a <b>n bytes</b> , anche se il numero di caratteri inseriti effettivamente sarà minore: quindi per n uguale a 50, questo campo occuperà 50 bytes <i>anche se inseriamo una quantità inferiore di caratteri</i> . Nota bene: per tutti i tipi CHAR, il valore di default di n è pari a 1.
<b>VARCHAR(n)</b>	Rappresenta una <b>stringa di lunghezza variabile</b> , con n indicante il massimo numero di caratteri memorizzabili (n, anche qui, deve essere sempre $\leq 8000$ ). A differenza di CHAR(n), <b>lo spazio occupato dipende dal numero effettivo di caratteri inseriti e non dalla specifica di n</b> .
<b>NCHAR(n)</b>	Rappresenta una <b>stringa testuale fissa di caratteri</b> equivalente al tipo CHAR(n) ma la codifica dei caratteri è <b>UNICODE</b> , per cui ogni carattere occupa 2 byte e <b>l'occupazione totale è <math>n * 2</math> bytes</b> . Per questo motivo il limite massimo per n è 4000 anziché 8000.
<b>NVARCHAR(n)</b>	Rappresenta una <b>stringa di lunghezza variabile</b> , equivalente al tipo NCHAR(n) ma la codifica dei caratteri è <b>UNICODE</b> , per cui ogni carattere occupa 2 byte e <b>l'occupazione totale è <math>n * 2</math> bytes</b> . Per questo motivo, anche per NVARCHAR(n), il limite massimo per n è 4000 anziché 8000.
<b>TEXT(n)</b>	Consente di memorizzare <b>testi a lunghezza variabile, ossia sequenze di caratteri particolarmente lunghe</b> (fino a una dimensione massima di 2 GigaByte), occupando non più di quanto richiesto dai caratteri effettivamente memorizzati.
<b>NTEXT(n)</b>	Consente di memorizzare <b>testi a lunghezza variabile, ossia sequenze di caratteri particolarmente lunghe</b> , occupando non più di quanto richiesto dai caratteri effettivamente memorizzati e utilizzando la codifica <b>UNICODE</b> . Il limite è quindi 1 GigaByte anziché 2.

### Dati di tipo Binari

Sono inclusi in questa categoria i tipi per memorizzare sequenze binarie generiche, utili ad archiviare documenti, immagini, video, suoni, oggetti e quanto altro codificato con sequenze più o meno lunghe di bit.

<b>BIT</b>	Rappresenta un <b>valore di tipo booleano</b> (vero o falso) e occupa <b>1 singolo bit</b> .
<b>BINARY(n)</b>	Rappresenta una <b>sequenza di bit di lunghezza fissa</b> , dove n indica il numero di byte da utilizzare (massimo 8000). L'occupazione in memoria è sempre pari a <b>n byte</b> .
<b>VARBINARY(n)</b>	Rappresenta una <b>sequenza di bit di lunghezza variabile</b> , dove n indica il numero massimo di byte utilizzabili ( $n \leq 8000$ ). L'occupazione in memoria è sempre pari al <b>numero effettivo dei byte di dati memorizzati + 2 byte</b> . Indicando <b>max</b> , al posto di n, si dispone di <b>(<math>2^{31}</math>)-1 byte</b> di spazio massimo.
<b>IMAGE(n)</b>	Rappresenta una <b>sequenza di bit di lunghezza variabile</b> , dove n indica il numero massimo di byte utilizzabili. A differenza di VARBINARY il <b>massimo valore ammesso per n è pari a 2.147.483.647</b> (ossia $2^{31} - 1$ ) per cui IMAGE si presta a contenere dati binari molto grandi come immagini, documenti, video, ecc.

**NOTA IMPORTANTE:** a partire da Sql Server 2005 sono state introdotte le versioni estese di tre tipi di dati: **VARCHAR(MAX)**, **NVARCHAR(MAX)**, **VARBINARY(MAX)** che consentono di archiviare grandi quantità di dati, superando il limite di 8000 fissato per le loro versioni originali e portandolo a  $2^{31} - 1$  pari a **2.147.483.647**. Hanno il grande pregio di poter sostituire i tipi di dato TEXT/IMAGE per la memorizzazione di dati molto grandi: quest'ultimi infatti sono deprecati, in quanto usano una tecnica di memorizzazione non conforme con gli altri dati.

### Dati di tipo Temporale

Si tratta dei tipi di dato usati in MS SQL Server per memorizzare e trattare date e ore.

<b>DATE:</b>	Rappresenta <b>una data</b> , archiviata come valore numerico e occupante una dimensione di <b>3 bytes</b> . Sono gestibili date comprese <b>fra il 1° gennaio 0001 d.c. e il 31 dicembre 9999 d.c.</b>
<b>DATETIME:</b>	Rappresenta <b>una data e un'ora</b> , archiviata come valore numerico e occupante una dimensione pari a <b>4 bytes</b> . Esiste un limite alle date trattabili: devono essere comprese <b>fra il 1° gennaio 1753 d.c. e il 31 dicembre 9999 d.c.</b>
<b>SMALLDATETIME:</b>	Rappresenta <b>una data e un'ora</b> , archiviata come valore numerico e occupante una dimensione pari a <b>2 bytes</b> . Esiste un limite alle date trattabili: devono essere comprese <b>fra il 1° gennaio 1900 d.c. e il 6 giugno 2079 d.c.</b>
<b>DATETIME2:</b>	Rappresenta <b>una data e un'ora</b> , archiviata come valore numerico e occupante una dimensione pari a <b>8 bytes</b> . Sono gestibili date comprese <b>fra il 1° gennaio 0001 d.c. e il 31 dicembre 9999 d.c.</b> e l'accuratezza dell'ora archiviata può arrivare a 100 nanosecondi.

### Altri Tipi di Dati

<b>ROWVERSION TIMESTAMP</b>	Rappresenta un <b>valore numerico che viene automaticamente incrementato ad ogni nuovo record inserito (o modificato)</b> . E' utile per generare automaticamente delle chiavi primarie. E' preferibile usare ROWVERSION poiché TIMESTAMP è ritenuto obsoleto.
<b>UNIQUEIDENTIFIER</b>	Rappresenta <b>una stringa binaria di numeri di 16 byte, generata automaticamente in modo tale da essere "univoca"</b> . Nel codice delle applicazioni esso viene gestito come tipo GUID ( <i>Globally Unique Identifier</i> ), largamente usato come identificativo univoco e per la definizione di chiavi e relazioni.
<b>Altri tipi</b>	Altri tipi interessanti sono il tipo <b>TABLE</b> (per contenere set di record per una successiva elaborazione), il tipo <b>HIERARCHYID</b> (per gestire elementi di un albero, quindi in una struttura gerarchica), il tipo <b>XML</b> (per i dati XML), il tipo <b>Sql_Variant</b> (per creare un campo capace di contenere tipi diversi di dato, anziché uno stesso tipo in tutte le righe), il tipo <b>CURSOR</b> (per la gestione dei cosiddetti "cursori"), ecc.