

معرفی انواع داده در SQL Server

آشنایی با انواع داده در SQL Server یکی از موارد پراهمیت و بنیادی در یادگیری پایگاه داده است. شما به عنوان فردی که با **پایگاه داده** کار می‌کند، چه به صورت حرفه‌ای و در سطوح سازمانی و چه به شکل مبتدی، باید دیتا تایپ‌ها و مفاهیم مربوط به آن‌ها را به خوبی درک کنید. به کمک نوع های داده، مواردی همچون افزایش دقت داده‌ها، بهینه‌سازی کوئری‌ها و به طور کلی، کیفیت داده‌ها تضمین می‌شود. اگر با انواع Data Type آشنایی ندارید، پیشنهاد می‌شود با خواندن این مطلب، دانش خود در حوزه پایگاه داده را افزایش دهید. البته پیش از بررسی انواع داده در SQL Server، قصد داریم مقدماتی از چیستی داده‌ها و نکات آن را شرح دهیم.

داده چیست؟

در SQL Server، داده‌ها به اطلاعاتی گفته می‌شود که با ساختار خاصی درون یک پایگاه داده ذخیره شده‌اند. با کمک **SQL Server** می‌توان داده‌های ذخیره‌شده در دیتابیس را مدیریت و استخراج کرد.

نکات مهم داده ها در SQL Server

نکات پراهمیت مربوط به انواع داده در SQL Server به شرح زیر است:

- **جدول‌ها:** هر یک از جداول پایگاه داده به اطلاعاتی خاصی اختصاص داده می‌شوند. به عنوان مثال، برای ذخیره‌سازی جزئیات مشتریان، یک جدول تحت عنوان Customers و برای نگهداری اطلاعات مربوط به محصولات، جدول Products را داریم.
- **سطرها (Rows):** درون هر یک از جداول پایگاه داده، سطرها (یا همان ردیف‌ها) قرار دارند که در آن‌ها داده‌های مرتبط به هم وجود دارد. به عنوان مثال، در جدول مشتری، هر یک از سطرها نمایانگر یک مشتری مختلف به همراه نام، آدرس و اطلاعات مربوط به ارتباط با آن‌ها است.
- **ستون‌ها (Columns):** با استفاده از ستون‌های پایگاه داده تعریف می‌شود که در هر سطر، چه نوع اطلاعاتی ذخیره خواهد شد. به عنوان مثال، ممکن است در جدول مشتریان (Customers)، ستون‌هایی با نام‌های شماره کاربری مشتری (CustomerId)، نام (FirstName)، نام خانوادگی (LastName) و موارد دیگر وجود داشته باشد.
- **نوع‌های داده (Data Types):** در SQL Server داده‌ها می‌توانند نوع داده مشخصی داشته باشند که این دیتا تایپ‌ها نشان می‌دهند داده‌ها به چه صورتی در هر یک از ستون‌ها ذخیره‌سازی شوند. به عنوان نمونه، ستون مربوط به تاریخ تولد (DateOfBirth) باید به صورت نوع داده تاریخ (Date) باشد.
- **کوئری‌ها (Queries | پرسمان‌ها):** به واسطه کوئری‌های SQL Server، این امکان برای شما فراهم می‌شود که سؤال‌های خاصی از دیتابیس بپرسید یا بخش‌های مشخصی از داده‌ها را استخراج کنید.

در این بخش، به مقدماتی از داده‌ها در SQL پرداختیم. حال در ادامه قصد داریم انواع داده در SQL Server را به طور مفصل مورد بررسی قرار دهیم.

انواع داده در SQL Server

در این بخش، انواع داده در SQL Server را معرفی خواهیم کرد. به واسطه دیتا تایپ‌ها می‌توان جنس یا همان نوع داده‌های ذخیره‌شونده در ستون‌های جداول را تعیین کرد. به بیان ساده، لازم است هر یک از ستون‌های یک جدول حاوی یک نوع داده مشخص باشند تا به این طریق مشخص شود آن ستون چه نوع‌هایی از Data را می‌تواند نگهداری کند. انواع داده در SQL Server شامل اعداد، متن، تاریخ و سایر موارد است که در ادامه، آن‌ها را طبقه‌بندی می‌کنیم.



نوع داده عددی دقیق (Exact Numeric)

از میان انواع داده در SQL Server، نوع داده‌های عددی Exact برای ذخیره‌سازی داده‌ها همراه دقت، مورد استفاده قرار می‌گیرند. این نوع‌های داده برای شرایطی مناسب‌اند که می‌خواهیم دقت داده‌ها به‌طور کامل حفظ شود. به‌عنوان مثال، وقتی که با داده‌های مالی، کمیت‌های دقیق یا شمارش آیتم‌ها سروکار داریم، انواع داده عددی دقیق کاربرد دارند. آن عبارتند از:

- **BIT:** بیت نشان‌دهنده یک مقدار دودویی با ۲ حالت ممکن (صفر و یک) است و معمولاً برای داده‌های بولین (Boolean) کاربرد دارد.

- **INT:** برای ذخیره‌سازی اعداد صحیح علامت‌دار (۳۲ بیتی) از نوع داده INT استفاده می‌شود. این دیتا تایپ بازه گسترده‌ای از مقادیر را پوشش می‌دهد. توجه کنید که معمولاً در اغلب موارد از این نوع داده برای نگهداری اعداد صحیح استفاده می‌شود.
- **TINYINT:** به‌منظور ذخیره‌سازی اعداد صحیح بدون علامت در رنج عددی صفر الی ۲۵۵ کاربرد دارد. معمولاً برای اعداد کوچک، مانند سن افراد یا تعداد موجودی یک محصول، این نوع داده به کار می‌رود.
- **SMALLINT:** اگر بخواهید اعداد صحیح بزرگ، مانند ID کارمندان یا تعداد سفارش‌ها را ذخیره کنید، نوع داده SMALLINT مناسب است.
- **BIGINT:** از میان انواع داده در SQL Server، این نوع Data Type به شما امکان نگهداری اعداد بسیار بزرگ را می‌دهد.
- **NUMERIC یا DECIMAL:** در شرایطی که بخواهید اعداد ممیز ثابت (با دقت مشخص) را ذخیره کنید، این نوع داده برای شما مطلوب است. به‌عنوان مثال، می‌توانید مقادیر ارزی یا محاسبات مالی همراه با دقت مشخص را با این Data Type در پایگاه داده خود نگهداری کنید.

Notes	Type	Storage	Max	Min	Datatype
	Exact	8 bytes	$2^{63}-1$	-2^{63}	Bigint
	Exact	4 bytes	2,147,483,647	-2,147,483,648	Int
	Exact	2 bytes	32,768	-32,768	Smallint
	Exact	1 bytes	255	0	Tinyint
	Exact	1 to 8 bit columns in the same table requires a total of 1 byte, 9 to 16 bits = 2 bytes, etc...	1	0	Bit
The Decimal and the Numeric data type is exactly the same. Precision is the total number of digits. Scale is the number of decimals. For both the minimum is 1 and the maximum is 38.	Exact	Precision 1-9=5 bytes, precision 10-19 =9 bytes, precision 20-28=13 bytes, precision 29-38=17 bytes	$10^{38}-1$	$-10^{38}+1$	Decimal
	Exact	same as Decmal	same as Decmal	same as Decmal	Numeric

نوع داده عددی تقریبی (Approximate Numeric)

در بین انواع داده در SQL Server، یک دیتا تایپ خاص برای ذخیره سازی اعداد ممیز شناور (Floating Point) و مقادیر عددی تقریبی وجود دارد. نوع داده های تقریبی برای مواردی به کار می روند که دقت (Precision)، یک معیار ضروری نیست و رند شدن (Rounding) یا تقریب زدن اعداد تا حدی قابل قبول باشد.

- **Float**: برای ذخیره سازی داده های عددی تقریبی (Approximate Numeric) که دقت (Precision) در آن ها متفاوت است، از دیتا تایپ فلوت استفاده می کنیم. به عنوان مثال، در محاسبات مهندسی، داده های علمی و سنجش دما می توان از Float استفاده کرد.
- **Real**: این DataType برای ذخیره سازی داده های ممیز شناور (با دقت واحد) استفاده می شود. مشابه نوع داده Float، در مواردی که با داده های عددی تقریبی سروکار داریم، آن را به کار می بریم.
- **DOUBLE PRECISION**: از میان انواع داده در SQL Server، این نوع داده از Float و Real دقت بیشتری دارد؛ اما همچنان در گروه ذخیره سازی داده های ممیز شناور دسته بندی می شود.

Notes	Type	Storage	Max	Min	Datatype
Precision is specified from 1 to 53.	Approx	4 bytes when precision is less than 25 and 8 bytes when precision is 25 through 53	1.79E+308	-1.79E + 308	Float
Precision is fixed to 7.	Approx	4 bytes	3.40E + 38	-3.40E + 38	Real

نوع داده رشته کاراکتری (Character String)

یکی دیگر از انواع داده در SQL Server، رشته‌های کاراکتری هستند که به منظور ذخیره‌سازی دیتای متنی و کاراکترها، مورد استفاده قرار می‌گیرند. این Data Type ها برای نگهداری تمامی نام‌ها، آدرس‌ها و اساساً توضیحات و محتوای متنی، ضروری محسوب می‌شوند. رایج‌ترین رشته‌های کاراکتری به شرح زیر است:

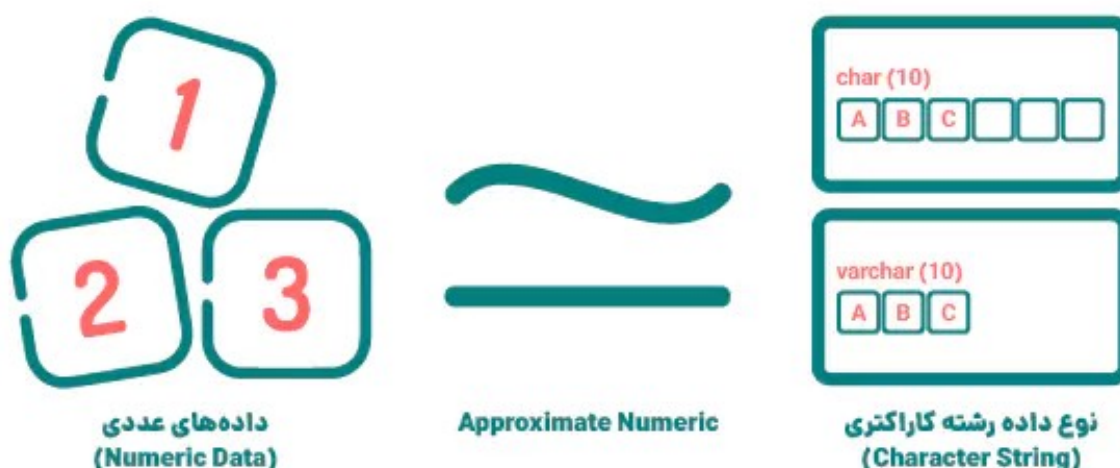
- **CHAR(n)**: برای ذخیره کردن مقادیری مانند کد پستی یا کدهایی که طولشان مشخص و ثابت است، استفاده می‌شود. در حقیقت، این نوع داده برای رشته‌های کاراکتری با طول ثابت (n) کاربرد دارد.
- **VARCHAR(n)**: اگر بخواهید یک داده متنی با «طول متغیر»، مثل نام افراد یا توضیحات، ذخیره کنید، نوع داده VARCHAR مناسب است. در چنین شرایطی متغیر n، نماد حداکثر طول آن خواهد بود.
- **TEXT**: ذخیره‌سازی محتوای متنی وسیع مانند مقالات، کامنت‌ها و یادداشت‌ها، همگی از طریق نوع داده Text امکان‌پذیر است.
- **NCHAR(n)**: در صورتی که نیاز باشد رشته‌های کاراکتری یونی کد (Unicode) با طول ثابت n ذخیره کنیم، این دیتا تایپ خاص از انواع داده در SQL Server مطلوب خواهد بود.
- **NVARCHAR(n)**: اگر بخواهید رشته‌های کاراکتری یونی کد با طول متغیر (حداکثر طول n) را ذخیره‌سازی کنید، باید از NVARCHAR(n) استفاده بشود.
- **NTEXT**: به منظور نگهداری محتوا یونی کد گسترده از این نوع داده کمک می‌گیریم.

Notes	Type	Storage	Max	Min	Datatype
Fixed width	String	Defined width	8000 char	0 char	Char
Variable width	String	2 bytes + number of chars	8000 char	0 char	Varchar
Variable width	String	2 bytes + number of chars	2 ³¹ chars	0 char	Varchar(max)
Variable width	String	4 bytes + number of chars	2,147,483,647 chars	0 char	Text
Fixed width	Unicode	Defined width x 2	4000 chars	0 char	Nchar
Variable width	Unicode		4000 chars	0char	Nvarchar
Variable width	Unicode		2 ³⁰ chars	0 char	Nvarchar(max)
Variable width	Unicode		1,073,741,823 chars	0 char	Ntext

nikamooz;
تجربه، آموزش، آینده

به طور کلی، انواع رشته‌های کاراکتری، چه آن‌هایی که یونی کد نیستند (CHAR, VARCHAR, TEXT) و چه آن‌هایی که یونی کد (NCHAR, NVARCHAR, NTEXT) هستند، همگی برای نگهداری اطلاعات متنی و مجموعه کاراکترها استفاده می‌شوند. معمولاً داده‌های یونی کد از چندین زبان مختلف یا مجموع کاراکترها پشتیبانی می‌کنند. بنابراین، هنگام انتخاب دیتا تایپ رشته‌ای از انواع داده در SQL Server، سعی کنید آن را براساس طبیعت داده‌هایی که می‌خواهید ذخیره کنید و طول متن مورد انتظار انتخاب کنید. استفاده از Data Type مناسب کمک می‌کند تا فضای ذخیره‌سازی بهینه‌سازی شود و این تضمین به شما داده می‌شود

که اسکیمای پایگاه داده دقیقاً منعکس کننده نیازمندی‌های داده‌ها سیستم شما باشد.



نوع داده دودویی (Binary)

دیتا تایپ دودویی (Binary) برای ذخیره‌سازی داده‌ها باینری استفاده می‌شود. این اطلاعات باینری مواردی همچون تصاویر، فایل‌های صوتی، داده‌های رمزگذاری شده و هرگونه از انواع داده در SQL Server که در قالب‌های عددی یا کاراکتری قرار نمی‌گیرد را پوشش می‌دهد. در ادامه، مرسوم‌ترین انواع داده باینری SQL Server را شرح می‌دهیم:

- **BINARY(n)**: برای ذخیره کردن داده‌های دودویی با طول ثابت (n) به کار می‌رود. به عنوان مثال، برای نگهداری مقادیر مربوط به کلیدهای رمزنگاری، این نوع داده مناسب است.
- **VARBINARY(n)**: از این نوع داده برای ذخیره‌سازی داده‌های دودویی با طول متغیر (حداکثر n)، مانند تصاویر و یا ضمیمه‌های اسناد، استفاده می‌شود.

- **IMAGE**: این نوع Data Type، امکان ذخیره‌سازی داده‌ها باینری به طول بسیار زیاد، شامل تصاویر، فایل‌های صوتی و اسناد باینری را به شما می‌دهد.
- **VARBINARY(max)**: مشابه VARBINARY عمل می‌کند اما محدودیت حداکثر طول را ندارد. این نوع داده در شرایطی کارایی دارد که طول داده‌های دودویی از قبل تعیین نشده است.

Notes	Type	Storage	Max	Min	Datatype
Fixed width	Binary		8000 bytes	0 bytes	Binary
Variable width	Binary		8000 bytes	0 bytes	Varbinary
Variable width	Binary		2 ³¹ bytes	0 bytes	Varbinary(max)
Variable width. Prefer to use the varbinary(max) type as the image type will be removed in future versions.	Binary		2,147,483,647 bytes	0 bytes	Image

نوع داده تاریخ (Date) و زمان (Time)

اگر بخواهید تاریخ و اطلاعات مرتبط با زمان را ذخیره کنید، از بین انواع داده در SQL Server، این دیتا تایپ‌ها به کار شما می‌آیند. نوع داده‌های Date و Time برای اپلیکشن‌های مختلفی، مانند پیگیری رویدادها، زمان‌بندی و برنامه‌ریزی و رسیدگی به داده‌های حساس به زمان، ضروری هستند. در ادامه انواع Data Type های تاریخ و زمان را معرفی خواهیم کرد.



نوع داده دودویی
(Binary)



نوع داده تاریخ (Date) و زمان (Time)

- **DATE:** در این نوع داده، تنها بخش «تاریخ» از اطلاعات زمانی ذخیره می‌شود و قالب آن به صورت YYYY-MM-DD است.
- **TIME:** این دیتا تایپ فقط قسمت زمان از هر اطلاعات تاریخی را نگهداری می‌کند. فرمت TIME به شکل HH:MM:SS.ssssss است که در آن، HH نمایانگر ساعت تلقی می‌شود.
- **DATETIME:** از میان انواع داده در SQL Server، این نوع داده هر دو اطلاعات تاریخ و زمان را ذخیره می‌کند. به عنوان مثال، این دیتا تایپ امکان نگهداری مقدار '۰۷-۱۰-۲۰۲۳' را دارد.
- **DATETIME2:** این Data Type در واقع نوع تعمیم داده شده DATETIME محسوب می‌شود و به شما این امکان را می‌دهد که برای بخش ثانیه، دقت بیشتری را ذخیره‌سازی کنید. به عنوان مثال، '۰۷-۱۰-۲۰۲۳ ۱۴:۳۰:۰۰.۱۲۳۴۵۶۷' یک نمونه از فرمت قابل نگهداری توسط DATETIME2 است.
- **SMALLDATETIME:** به واسطه این نوع داده، می‌توانید تاریخ و زمان را ذخیره کنید؛ اما این دیتا تایپ در مقایسه با DATETIME، بازه محدودتری را پوشش می‌دهد و دارای دقت کمتری است. فرمت کلی آن به صورت 'YYYY-MM-DD HH:MM:SS' است.
- **TIMESTAMP:** همان‌طور که از نام آن قابل درک است، از انواع داده در SQL Server، نوع تایم استمپ به تاریخ و زمان مرتبط نیست. در حقیقت، TIMESTAMP به منظور تعیین نسخه و پیگیری تغییرات اعمال شده روی یک سطر جدول به کار می‌رود؛ به گونه‌ای که با ویرایش یک سطر، ستون‌های TIMESTAMP به صورت خودکار توسط SQL Server به‌روزرسانی می‌شوند تا بدین طریق بتوانید روند تغییرات را در دسترس داشت باشید.
- **DATETIMEOFFSET:** در بین انواع داده در SQL Server، عملکرد این نوع دیتا تایپ نیز مشابه DATETIME است؛ با این تفاوت که اطلاعات مربوط به Time Zone را نیز نگهداری می‌کند. قالب ساختاری این نوع داده به صورت 'YYYY-MM-DD HH:MM:SS.ssssss -hh:mm' طراحی شده که در آن '-hh:mm' بیانگر آفست منطقه زمانی یا همان Time Zone Offset است.

Notes	Type	Storage	Max	Min	Datatype
If you are running SQL Server 2008 or later and need milliseconds precision, use datetime2(3) instead to save 1 byte.	Datetime	8 bytes	9999-12-31 23:59:59.997	1753-01-01 00:00:00.000	Datetime
	Datetime		2079-06-06 23:59	1900-01-01 00:00	Smalldatetime
	Datetime	3 bytes	9999-12-31	0001-01-01	Date
Specifying the precision is possible. TIME(3) will have milliseconds precision TIME(7) is the highest and the default precision. Casting values to a lower precision will round the value.	Datetime		23:59.9999 999	00:00:00.00 00000	Time
Combines the date datatype and the time datatype into one. The precision logic is the same as for the time data type.	Datetime	Precision 1-2 =6 bytes precision 3-4 =7 bytes precision 5-7 =8 bytes	9999-12-31 23:59:59.99 99999	0001-01-01 00:00:00.00 00000	Datetime2
Is a date time2 datatype with the UTC offset appended.	Datetime	Precision 1-2 =8 bytes precision 3-4 =9 bytes precision 5-7 =10 bytes	9999-12-31 23:59:59.99 99999 +14:00	0001-01-01 00:00:00.00 00000-14:00	Datetimeoffset
Stores a database-wide unique number that gets updated every time a row gets updated.	Other				Timestamp

نوع داده تعریف شده توسط کاربر (User Defined)

یکی دیگر از دیتا تایپ‌های پراهمیت که در این بخش مقاله انواع داده در SQL Server بررسی می‌کنیم، نوع داده تعریف شده توسط کاربر (UDDT) است. دیتا تایپ User Defined، نوع داده دلخواه و مشخصی است که توسط کاربر یا برنامه‌نویس پایگاه داده ایجاد می‌شود و به واسطه آن، مجموعه قوانین یا محدودیت‌های خاصی برای داده‌های قابل ذخیره‌سازی در یک ستون تعریف خواهد شد. نوع داده تعریف شده توسط کاربر در مواردی همچون تسهیل طراحی پایگاه داده، بهبود سازگاری، افزایش خوانایی و نگهداری Schema های دیتابیس کارآمد به حساب می‌آید.

نوع داده CLR

نوع داده CLR یک دیتا تایپ دلخواه است که با استفاده از **Common Language Runtime فریم ورک .NET** تعریف می‌شود. این نوع از انواع داده در SQL Server با گسترش داده‌های اصلی خود SQL Server ایجاد می‌شوند و می‌توان از آن‌ها در کپسوله‌سازی منطق و رفتار پیچیده درون یک دیتا تایپ استفاده کرد. به طور خاص، نوع داده CLR برای اجرای محاسبات پیچیده یا پیاده‌سازی منطق تجاری دلخواه درون پایگاه داده مناسب است.

سایر نوع های داده

علاوه بر انواع داده در SQL Server که در این بخش به آن‌ها اشاره کردیم، این سیستم مدیریت پایگاه داده تعدادی دیتا تایپ خاص دارد که در برخی سناریوها استفاده می‌شوند.

- **UNIQUEIDENTIFIER**: به واسطه این نوع داده، می‌توان مقدار یک Unique Identifier را به صورت سراسری (Global) تعریف کرد. معمولاً از آن به عنوان کلید اصلی یا روشی استفاده می‌شود که بتوان سطرهای پایگاه داده را به صورت یکتا مشخص کرد. این موضوع به طور ویژه در زمان‌هایی اهمیت دارد که به همگام‌سازی داده‌ها در چندین دیتابیس یا سیستم نیاز داریم.
- **XML**: برای ذخیره‌سازی داده‌های **زبان نشانه گذاری گسترش پذیر** (Extensible Markup Language) در یک قالب ساختارمند، از نوع داده XML در SQL Server استفاده می‌شود. XML یک استاندارد مشهور برای نمایش داده‌های ساختار یافته و نیمه ساختاریافته (Semi-Structured) محسوب می‌شود و استفاده از زمانی مناسب است که باید داده‌ها به صورت سلسله مراتبی (Hierarchical) یا تودرتو (Nested) ذخیره شوند. در SQL Server داده‌های XML پشتیبانی می‌شوند و امکان دست‌کاری، کوئری‌نویسی و ذخیره‌سازی داده‌های XML به صورت کارآمد فراهم شده است.
- **JSON**: در SQL Server داده‌های JSON پشتیبانی می‌شوند و شما می‌توانید این نوع دیتا را به صورت کارآمد ذخیره کنید و به کوئری‌نویسی با آن‌ها بپردازید. JSON داده‌های سبک و پراستفاده‌ای هستند و وقتی بخواهیم به ذخیره‌سازی و پردازش داده‌های نیمه ساختاریافته و ساختاریافته بپردازیم، مفید واقع خواهند شد.
- **GEOMETRY and GEOGRAPHY**: این انواع داده در SQL Server برای کار با داده‌های فضایی کارایی دارند و به کمک آن‌ها، می‌توان اشکال جغرافیایی و هندسی را دست‌کاری و ذخیره‌سازی کرد. داده‌های فضایی در اپلیکیشن‌هایی کارایی دارند که مواردی مانند نگاشت (Mapping)، سیستم‌های اطلاعاتی جغرافیایی (GIS) و سرویس‌های مبتنی بر لوکیشن را شامل می‌شوند. از این رو، استفاده از چنین دیتا تایپ‌هایی از میان انواع داده در SQL Server توصیه می‌شود.
- **SQL_VARIANT**: نوع خاصی از انواع داده در SQL Server به شمار می‌رود که به واسطه آن می‌توان انواع مختلفی از دیتا را در یک ستون واحد، متغیر یا پارامتر ذخیره کرد. SQL_VARIANT به منظور ایجاد انعطاف در مواقعی طراحی شده است که نیاز باشد با انواع مختلفی از داده‌ها درون یک فیلد کار کنیم. البته توجه کنید که این دیتا تایپ باید با دقت و احتیاط مورد استفاده قرار گیرد؛ چون چالش‌های مختلفی درخصوص کارایی کوئری‌ها و تبدیل نوع داده با آن همراه است.

Notes	Type	Storage	Max	Min	Datatype
Stores value of various SQL Server-supported data types, except text, next, and timestamp.	Other				Sql-variant
Stores XML data. You can store xml instances in a column or a variable.	Other				Xml
Stores a globally unique identifier (GUID).	Other				Uniqueidentifier

کلام آخر: انواع داده در SQL Server کدامند؟

در این مقاله به معرفی انواع داده در SQL Server پرداختیم تا به این طریق بتوانید آن را به عنوان راهنما در انتخاب دیتا تایپ های ستون های Database مورد استفاده قرار دهید. اساساً انتخاب نوع داده مناسب برای ستون های پایگاه داده امری پراهمیت است؛ چون از این راه اطمینان خواهید داشت که مواردی مانند یکپارچگی داده ها، کارایی در ذخیره سازی و بهینه شدن کوئری ها تضمین می شوند. کافیست نوع داده ها را براساس خصوصیات ذاتی داده هایی تعیین کنید که در آن ستون ذخیره خواهند شد.