**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**انواع داده ها**

همانطور که می دانیم Data type ها مبحث گسترده ای دارند. در SQL Server هر ستون ، متغیرهای محلی یا local Variable ها ، عبارت ها و پارامتر ها ، نوع داده یا Data type مربوط به خود را دارند.

**Datatype ها در اس کیو ال سرور به 7 گروه ذیل تقسیم می شوند:**

1. Exact numeric: مقادیر عددی دقیق

2. Approximate numeric: مقادیر عددی تقریبی

3. Date and time: تاریخ و ساعت

4. Character strings: رشته های کاراکتری

5. Unicode character strings: رشته های کاراکتری یونیکدی (مناسب برای فارسی)

6. Binary strings: رشته های باینری

7. Other data types: سایر انواع داده

#### **انواع داده های character String یا رشته کاراکتر**

* **Char (n):** این نوع داده دارای طول ثابتی بوده و n بایت داده در آن بصورت Non-Unicode نگهداری می شود) n اندازه ذخیره سازی است (.N می تواند از 1 تا 8000 باشد. اگر در ستونی مقدار فرضا 25 کاراکتر را برای charدر نظر بگیریم و داده ی وارد شده کمتر از این مقدار باشد. فضای باقی مانده با Space از سمت راست پر خواهد شد.که به هنگام رویت داده، این فضا نادیده گرفته می شود.
* **Varchar(n):**دارای طول متغیر بوده و کاراکترهای آن بصورت Non-Unicode هستند. و می توانند ارزشی از 1 تا 8000 داشته باشند. داده ی وارد شده می تواند 0 کاراکتری نیز باشد. اندازه ذخیره سازی این نوع داده ، اندازه ی طول واقعی اطلاعات وارد شده + 2 بایت است. در varchar برخلاف char داده ی وارد شده تنها به اندازه کاراکترهای وارد شده فضا اشغال می کنندو می توان حداکثر طول کاراکتر را مشخص کرد به عنوان مثال varchar(6) نشان می دهد که این نوع داده شامل 6 کاراکتر بوده و حداکثر می تواند شش کاراکتر را ذخیره کند.
* Varchar(max): طول متغیر داشته و کاراکتر های آن به صورت non-Unicode هستند. حداکثر حجم ذخیره سازی این نوع داده 1-31^2یا(2,147,483,647) می باشد (2GB). داده ی وارد شده می تواند null یا 0 کاراکتر باشد. در اینجا نیز اندازه ذخیره سازی ، اندازه ی واقعی داده های وارد شده +2 بایت است. به اینگونه نوع داده ها large-value data type می گویند.
* **Text:** نوع داده ی text نیز دارای طول متغیر بوده و طول رشته ی آن حداکثر 31^2 یا (2,147,483,647) و بصورت non-Unicode در code page سرور می باشد. وقتی که code page از کاراکتر های دو بایتی استفاده می کند ، حافظه همچنان 2,147,483,647 است. بسته به نوع رشته کارکتر ها اندازه حافظه و ذخیره سازی ممکن است کمتر از 2,147,483,647 بایت باشد. از این نوع داده برای ذخیره ی داده هایی با بیش از 8000 کاراکتر استفاده می کنند.

#### **نوع داده های Unicode String یا رشته یونیکد**

* **Nchar(n):**طول ثابتی دارد و بایت داده در آن بصورت Unicode نگهداری می شوند، اندازه ذخیره سازی آن [2\*n]  است وچون هر کاراکتر Unicode برای ذخیره سازی به دو بایت نیاز دارد،حداکثر اندازه ی این نوع 4000 کارکتر است بنابراین n ارزشی بین 1 تا 4000 را در بر می گیرد.داده ها می توانند رشته هایی تک بایتی یا چند بایتی از حروف ،ارقام و سایر سمبل هایی باشند که توسط مجموعه کدهای موجود در database locale پشتیبانی شده ان. (database locale محلی است که مجموعه ای از زبان ها، مناطق و کشورها و نیز کدهایی که پایگاه داده برای تفسیر صحیح انواع داده (nvarchar - nchar) را در یک پایگاه داده ویژه مشخص می کند). همچنین nchar تعداد محدود و مشخصی از کاراکترها را در خود جای می دهد مثلا nchar(30) همیشه 30 کاراکتر را ذخیره می کند حتی اگر رشته ای کوتاهتر از 30 کاراکتر به آن اختصاص یابد.
* **Nvarchar(n):**داده ها با طول متغیر در ستون ذخیره می شوند و Unicode می باشند. و n می تواند ارزشی بین 1 تا 4000 را شامل شود. داده ی وارد شده می تواند صفر یا Null نیز باشد . اندازه ی ذخیره سازی بر حسب بایت ، دو برابر تعداد کاراکترهای وارد شده +2 بایت است.
* **Nvarchar(max):**طول داده متغیر و به صورت Unicode است حداکثر حجم ذخیره سازی این نوع داده 1-31^2 یا (2,147,483,647) می باشد . (2GB) همچنین داده ی وارد شده می تواند صفر یا Null باشد و اندازه ی ذخیره سازی بر حسب بایت ، دو برابر تعداد کاراکترهای وارد شده +2 است. Navarchar (max) یک large-value data type محسوب می شود.
* **Ntext:** داده در این نوع طول متغیر داشته و Unicode می باشد. و حداکثر طول رشته آن 1-30^2 یا (1,073,741,823) بایت است . حجم ذخیره سازی نیز دو برابر طول رشته ی وارد شده و بر حسب بایت است. معمولا از این نوع داده برای داده هایی با بیش از 8000 کاراکتر استفاده می کنند.

#### **انواع داده ی Binary String رشته دودویی**

* **Binary(n):**دارای طول ثابت بوده و n میتواند مقادیر 1 تا 8000 بایت را در بر گیرد. حجم حافظه و ذخیره سازی در آن نیز n است. از باینری زمانی استفاده می شود که طول داده های وارد شده در ستون سازگار باشند.
* **Varbinary(n|max):**داده های باینری طول متغیر دارند. N می تواند مقادیر 1 تا 8000 بایت را شامل شود. **Max** نیز بیانگر این مطلب است که حداکثر حجم حافظه 1-31^2 بایت است. اندازه ذخیره سازی نیز برابر طول واقعی داده ی وارد شده + 2 بایت است. داده وارد شده می تواند ارزش Null و طول 0 بایت را داشته باشد. عبارت مترادفی که ANSI SQL برای varbinary در نظر گرفته binary varing است. Varbinary هنگامی بکار می رود که اندازه داده های وارد شده در ستون ها بطور قابل توجهی متفاوت باشند. و از varbinary max نیز زمانی استفاده می شود که داده های وارد شده در ستون بیش از 8000 بایت باشند.
* **Image:** داده های باینری با طول متغیر از 0 تا 1-31^2 یا (2,147,483,647) را شامل می شود. برای ذخیره تصاویر مثلا لوگوی یک شرکت از این نوع داده استفاده می شود. تصویر ذخیره شده بعنوان رشته ای از بیت ها ذخیره شده و توسط SQL تفسیر نمی شود. و تفسیر هر گونه داده ی تصویری ذخیره شده به کمک Application ها می باشد. مثلا یک Application می تواند یک داده ی موجود در Image column را با فرمت JPEG ذخیره کند. Application ای که داده ها را از Image column می خواند باید به درستی قادر به تشخیص فرمت داده ها و نمایش آنها باشد. یک Image column یک location برای ذخیره رشته بیتها که در واقع value ای را برای داده های تصویری می سازند، فراهم می کنند.

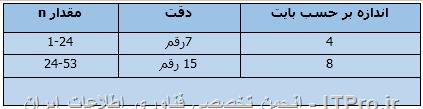
#### **انواع داده ی Exact Numeric یا داده عددی**

* **Int:** اندازه آن 4 بایت است و قادر به ذخیره سازی محدوده اعداد صحیح 31^2- یا (2,147,483,648-) تا 1-31^2یا (2,147,483,647). می باشد .مترادف int در SQL-92 کلمه ی integer می باشد.
* **Bigint:** قابلیت ذخیره سازی 8 بایت را دارا می باشد و محدوده اعداد صحیح 9,223,372,036,854,775,808- تا 9,223,372,036,854,775,807 را شامل شده و کاربرد آن هنگامی است که مقدار داده ی صحیح بیش از نوع داده ی int باشد.
* **Smallint:** قابلیت ذخیره 2 بایت داده ی صحیح را داراست و می تواند اعدادی بین 32,768- تا 32,767 را شامل شود.
* **Tinyint:** قادر به ذخیره تنها یک بایت می باشد و محدوده اعداد صحیح 0 تا 255 را در بر می گیرد.
* **Bit:** مقادیر صحیح 0 یا 1 را در بر می گیرد.و به عنوان مثال برای وارد کردن داده های چون Y//N به معنی بله و خیر یا T//F به معنی درست یا نادرست به کار می رود.
* **Numericو decimal:**این نوع داده ها برای ذخیره داده هایی با اعداد اعشاری دارای ممیز شناور بدون گرد کردن استفاده می شوند و باید در آنها کل ارقام قابل ذخیره (دقت) و ارقام اعشاری قرار گرفته در سمت راست ممیز اعشار مشخص می شود. این نوع داده ها محدود1-38^10- تا 1-38^10 را در بر می گیرند.
* **Money:** برای ذخیره مقادیر پول کاربرد دارد البته برای پولهایی مانند ریال کارایی چندانی ندارد. و به عنوان برای ذخیره داده ای با واحد دلار که پول خرد دارد مثلا 2 دلار و 75 سنت ،استفاده ازاین نوع داده نمود بهتری دارد. چرا که این نوع داده دقیقا 4 رقم در سمت راست ممیز اعشار خواهد داشت. اندازه ذخیره سازی آن 8 بایت بوده و مقادیر 922,337,203,685,477.5808- تا 922,337,203,685,477.5807 را شامل می شود . همچنین این نوع داده برای هر نوع ارزی بکار برده می شود.
* **Small money:** اندازه آن 4 بایت است و مقادیر ارزی موجود در محدوده 214,748.3648- تا 214,748.3647 را می تواند ذخیره کند. این نوع داده دقیقا 4 رقم در سمت راست ممیز اعشار خواهد داشت و برای برای هر نوع ارزی کاربرد دارد.

#### **انواع داده Approximate Numeric یا تقریبا عددی**

اگر داده های با ممیز شناور با دقت لازم در فرمت باینری سرور قابل ارائه نباشند، SQL Server آنها را گرد می کند. وقتی که انواع داده با ممیز شناور را در نظر بگیریم ، در واقع دقت را برای حداکثر تعداد ارقام مجاز در هر دو طرف ممیز اعشار مشخص کرده ایم . نوع داده ها ی real و float برای ذخیره ی داده هایی های مناسب هستند که دقت مطلق برای آنها ضروری نیست و داده ها ارقام بسیار کوچک و یا بسیار بزرگ را شامل می شوند. مثل داده های علمی و آماری.

* **float(n):** محدوده ی 1.79 E+308- تا 1.79 E +308 برای این Data type مشخص شده است. N تعداد بیت های استفاده شده برای ذخیره مقدار اعشاری یک float number در نماد علمی است در نتیجه دقت و حجم ذخیره سازی را مشخص می کند.N ارزشی در محدوده 1 تا 53 را دارا می باشد. فضای ذخیره سای این نوع داده 8 بایت است.



**Real:**این نوع داده محدوده اعداد اعشاری 3.40 E+38- تا 3.40 E + 38 را در بر می گیرد. همانطور که مشاهده می شود دقت این گونه اعداد تا 38 رقم می باشد . فضای لازم برای ذخیره سازی این نوع داده 4 بایت است . در SQL Server نو ع داده ی real مترادف با float(24) می باشد.

#### **انواع داده ی Date and Time یا زمان و ساعت**

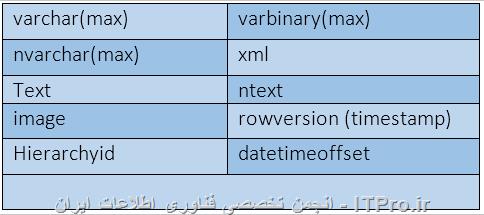
* **Date:** برای ذخیره ی تاریخ کاربرد دارد. فرمت رشته ای آن به صورت YYYY-MM-DD می باشد که YYYY محدوده ارقام 0001 تا 9999 را برای نمایش سال و MM ارقام 01 تا 12 را برای نمایش ماه و DD ارقام 01 تا 31 را با توجه به ماه مورد نظر ،برای نمایش روز می توانند در بر گیرند. مقدار پیش فرض این نوع داده 01-01-1900 است. صحت و درستی آن یک روز در نظر گرفته شده و اندازه حافظه مورد نیاز این نوع داده 3 بایت و ثابت است.
* **Time:** ستون های از این نوع زمان را در خود ذخیره می کنند. فرمت رشته ای آن hh:mm:ss[.nnnnnnn] می باشد و محدوده 00:00:00.0000000 تا 23:59:59.9999999 می تواند شامل شود. hh اعداد 0 تا 23 را برای نمایش ساعت و mm دو رقم از 0 تا 59 را برای دقیقه و ss عددی دو رقمی از 0 تا 59 را برای نمایش ثانیه می تواند در بر گیرد. n صفر تا هفت رقمی است که بیانگر کسر ثانیه می باشدو محدوده ارقام 0 تا 9999999 برای آن لحاظ شده است. دقت این نوع داده 100 نانو ثانیه می باشد و اندازه حافظه مورد نیاز برای ذخیره سازی آن 5 بایت و ثابت است .
* **Datetime2:** فرمت آن به صورت YYYY-MM-DD hh:mm:ss [.fractional seconds] می باشد و محدوده ی تاریخی 01-01-0001 تا 31-12-9999 را با مقدار پیش فرض 00:00:00 01-01-1900 شامل می شود n می تواند صفر تا هفت رقم در محدوده ی 0-9999999 را برای نمایش کسر اعشار در بر گیرد.دقت آن 100 نانو ثانیه می باشد. حافظه ی مورد نیاز برای ذخیره ی ارقام با دقت 3 مقدار 3 بایت برای دقت های 3 و 4 حدود 7 بایت و برای دقت های دیگر 8 بایت می باشد.
* **Datetime:** ستون هایی که این نوع داده برای آنها مشخص شده است می توانند تاریخ های January 1, 1753 تا December 31, 9999 و ساعت را از 00:00:00 تا 23:59:59.997 را در خود جای دهند. مقدار پیش فرض این داده در SQL سرور 00:00:00 01-01-1900 است.با توجه به فرمت در نظر گرفته شده ی hh:mm:ss [.nnnnnn] برای ساعت nمی تواند صفر تا سه رقم از محدوده اعداد 0 تا 999 را برای بخش کسر ثانیه شامل شود.دقت آن اعداد گرد شده ی رو به افزایش از 0.000 ، 0.003 یا 0.007 ثانیه می باشد.فضای لازم برای ذخیره سازی آن 8 بایت است.
* **Smalldatetime:** برای ذخیره تاریخ های01-01-1900 تا 06-06-2079 و نیز ساعت های 00:00:00 23:59:59 از smalldatetime با دقت یک دقیقه استفاده می شود. نکته ای که در این نوع داده مطرح می باشد این است که ثانیه هایی که از 29.998 و یا کمتر از آن هستند به نزدیکترین دقیقه ی قبلی و نیز ثانیه هایی که 29.999 و یا بیشتر از این مقدار هستند به نزدیکترین دقیقه ی بعد این مقدار گرد می شوند . به عنوان مثال ارزش 2007-05-09 23:59:59 به ارزش 2007-05-10 00:00:00 گرد و تبدیل شده است. مقدار پیش فرض در این نوع داده 00:00:00 01-01-1900 می باشد. حافظه ی مورد نیاز برای ذخیره سازی آن 4 بایت و ثابت است.
* **Datetimeoffset:** مقادیر 01-01-0001 تا 31-12-9999 را برای تاریخ و 00:00:00 تا 23:59:59.9999999 را برای ساعت شامل می شود .فرمت آن به شکل YYYY-MM-DD hh:mm:ss [.nnnnnn] می باشدو نیز برای جبران زمانی منطقه از فرمت [hh:mm {-//+}] در نظر گرفته شده است .باتوجه به بخش [.nnnnnnn] [{+//-}hh:mm.] مقدار n صفر تا هفت رقم در محدوده ی 0 تا 9999999 بوده و hh دو رقم در رنج -14 تا 14 و نیز mm دو رقم در محدوده ی 00 تا 59 را می تواند داشته باشد.محدوده جبران منطقه زمانی برای datetimeoffset از -14:00 تا +14:00 می باشد. دقت این نوع داده 100 نانو ثانیه بوده و 10 بایت حافظه برای ذخیره سازی به صورت ثابت نیاز دارد.

#### **دیگرData Type های**

**Cursor:** این نوع داده برای ذخیره ی متغیر ها یا پارامترهای OUTPUT مربوط به store proceduere کاربرد دارد.

می توان گفت که cursor به مجموعه ای از سطر ها اشاره دارد.هر متغیری که با این نوع داده ایجاد شود nullable می باشد. باید توجه داشت که این نوع داده نمی تواند بهنگام ایجاد یک table مورد استفاده قرار گیرد.

* **sql\_variant:** برای ذخیره سازی انواع داده های مختلف که SQL Server از آنها پشتیبانی می کند به کار می رود. این نوع داده در ستون ها ، پارامترها ،متغیر ها و مقادیر بازگشتی توابع تعریف شده توسط کاربر استفاده می شود. sql\_\_variant درواقع object های پایگاه داده را قادر می سازد تا datatype های دیگر را ساپورت کنند. یک ستون با نوع sql\_\_variant ممکن است شامل ردیف هایی با انواع داده ی متفاوت باشد به عنوان مثال برای ستونی که نوع داده ی sql\_\_variant برای آن تعریف شده می تواند مقادیر int یا binary و یا char را شامل شود. این نوع داده به عنوان مقدار پیش فرض نیز در نظر گرفته می شود. Sql\_\_variant حداکثر می تواند طولی برابر با 8016 بایت داشته باشد. در جدول زیر انواع داده هایی که بواسطه ی sql\_\_variant ذخیره نمی شوند ذکر شده است:



* **table:** نوع ویژه ای از انواع داده است که می تواند مجموعه ای از نتایج برای پردازش در زمانی دیگر ذخیره نماید.
* **rowversion,timetamp:** نوع داده ی rowversionعموما به عنوان مکانیزمی برای علامت گذاری ردیف های یک جدول استفاده می شود. حافظه ی مورد نیاز برای ذخیره سازی آن 8 بایت است این نوع داده فقط افزایش یک تعداد شماره را انجام می دهد و قادر به حفظ و نگهداری تاریخ یا ساعت نیست. بنابراین برای ثبت تاریخ یا ساعت از datetime2 استفاده می شود. هر database دارای یک شمارنده است که با درج داده و یا با اجرای عملیات به روزرسانی روی یک table که در بردارنده ی یک ستون از نوع rowversion در پایگاه داده است ، افزایش می یابد. هرجدول می تواند تنها یک ستون از نوع rowversion داشته باشد و هرزمانی که یک ردیف از یک ستون نوع rowversion تغییر یافته و یا در آن درج شود ، مقدار rowversion پایگاه داده افزایش یافته و در ستون موجود با این نوع داده وارد می شود. این ویژگی متغیر بودن باعث می شود که ستون با نوع داده ی rowversion گزینه ی مناسبی برای کلید ها به ویژه برای primary key ها نباشد. نوع داده ی timestamp نیز معادل با rowversion می باشد و از رفتار نوع داده ی مترادف خود تبعیت می کند. در DDL statement ها هر جا که ممکن است از rowversion به جای time stamp استفاده می شود.

**نکته :**DDL مخفف عبارت Data Definition Language بوده و دستورهای آن برای ایجاد ،تغییر و drop کردن ساختار داده ها در دریک Instance از SQL Server کاربرد دارند.

* **Uniqueidentifier:** یک ستون یا متغیر local با نوع داده ی uniqueidentifier می تواند به کمک دو روش زیر مقادیر اولیه را کسب کند: ا- به کمک تابع NEWID و 2- تبدیل یک رشته ی ثابت به فرم xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx که هر x یک عدد هگزادسیمال در محدوده ی 0 تا 9 یا a تا f را می تواند در بر داشته باشد. به عنوان مثال (6F9619FF-8B86-D011-B42D-00C04FC964FF) یک مقدار uniqueidentifier معتبر می باشد. همچنین عملگرهای مقایسه ای می توانند با مقادیر uniqueidentifier استفاده می شوند تنها عملیاتی که می توانند در برابر مقادیر uniqueidentifier صورت گیرند انجام مقایسه با کمک عملگر های (=, <>, <, >, <=, >=) و چک کردن null ویا notnull می باشد.
* **Xml:** این نوع داده برای ذخیره سازی داده های XML کاربرد دارد .به کمک این نوع داده می توانیم XML instance ها را در یک ستون داشته باشیم و یا یک متغیر از نوع XML را ذخیره کنیم. نمایش instance های ذخیره شده از نوع XML نمی تواند از اندازه 2GB تجاوز کند.
* **Hierarchyid:** یک نوع داده ی سیستمی با طول متغیر می باشد. از این نوع داده برای نمایش یک position در یک ساختار سلسله مرتبی استفاده می شود. ستون های تعریف شده از نوع hierarchid به صورت خودکار نشان دهنده ی یک tree نیستند و در واقع یک application است که به تولید و assign کردن مقادیر hierarchyid می پردازد به طوریکه رابطه مورد نظر میان ردیف ها در value ها مشخص شده باشد.

|  |
| --- |
| SELECT TypeName = name,TypeID = system\_type\_id FROM sys.types ORDER BY TypeID |