

نوع داده	میزان ذخیره سازی در حافظه
Bit	1 بیت
TinyInt	1 بایت
SmallInt	2 بایت
Int	4 بایت
BigInt	8 بایت
Numeric	5 الی 17 بایت
Decimal	5 الی 17 بایت
SmallMoney	4 بایت
Money	8 بایت
Real	4 بایت
Float(n)	4 یا 8 بایت
Char(n)	1 بایت برای هر کاراکتر، به تعداد n تا حداکثر 8000 بایت
VarChar(n)	1 بایت برای هر کاراکتر، به تعداد n تا حداکثر 8000 بایت
VarChar(max)	1 بایت برای هر کاراکتر، حداکثر 2 گیگابایت

NChar(n)	2 بایت برای هر کاراکتر به تعداد n تا حداکثر 4000 بایت
NVarChar(n)	2 بایت برای هر کاراکتر به تعداد n تا حداکثر 4000 بایت
SmllDateTime	4 بایت
DateTime	8 بایت
DateTime2(FSP)	6 تا 8 بایت
DateTimeOffset(FSP)	8 تا 10 بایت
Data	3 بایت
Time(FSP)	3 تا 5 بایت

متغیرهای enumeration و record و tuple از نوع reference type و متغیر struct از نوع value type می باشد.

تفاوت حافظه stack و heap

حافظه stack یک حافظه ثابت یا static ، محدود اما سریع است.

حافظه heap یک حافظه داینامیک با سایز متغیر اما کند تر از حافظه stack است.

حافظه stack از نوع **LIFO (Last input first out)** است (یعنی : اطلاعات پشت سر هم و به ترتیب قرار می گیرند به این صورت که آخرین داده ذخیره شده در بالای stack قرار می گیرد و به اصطلاح push می شود، حال اگر قصد برداشتن اطلاعات یا به اصطلاح pop کردن اطلاعات را داشته باشیم آخرین اطلاعات وارد شده در stack را در اختیار داریم) **در حالی که حافظه heap از نوع FIFO (First input first out)** است.

حافظه stack از نوع value type و حافظه heap از نوع reference type است.

این حافظه ها در قسمت user space حافظه قرار دارند.

حافظه stack توسط cpu به صورت خودکار مدیریت میشود، اما حافظه heap باید به صورت دستی توسط برنامه نویس مدیریت شود.

در نوع متغیر value type ، مقادیر تا اجرای یک متد در حافظه stack قرار دارند و پس از پایان اجرای متد، به صورت خودکار حذف میشوند.

اما در reference type ، عمر مقادیر منوط به یک متد یا region خاص نیست و تا زمانی که Garbage Collector این فضا را آزاد کند یا توسط برنامه نویس داده ها از حافظه heap پاک نشوند در حافظه باقی می ماند.