

## آموزش اصول متغیرها در SQL SERVER/SQL SERVER متغیرها در SQL SERVER

### آموزش اصول متغیرها در SQL SERVER

میدانیم که بعضی از مقادیر مثل 242 یا james kningt مقادیری ثابت هستند چون قبل و بعد از اجرای برنامه دست نخورده باقی میمانند.

اگر از یک نوع مقدار به طور مرتب استفاده میکنید، میتوانید مقداری حافظه را به آن اختصاص دهید. این به شما اجازه میدهد که مقدار را در حافظه ذخیره کرده و هر بار که به آن نیاز داشتید آن را تغییر دهید.

برای این کار، **sql interpreter** به دو قطعه اطلاعات نیاز دارد. اسم، و مقدار فضایی مورد نیاز برای ذخیره ی مقدار

تمرین: معرفی متغیرها

1. وارد کامپیوتر شوید

2. وارد **microsoft sql server** شده و بر روی **connect** کلیک کنید.

3. در **standard toolbar**، بر روی **new query** کلیک کنید

مشخص کردن متغیر

متغیر، مقداری از حافظه است که مقادیر مورد استفاده ی برنامه را روی آن ذخیره میکند. قبل از استفاده از متغیر **interpreter** باید اطلاع داشته باشید. این کار مشخص کردن متغیر نامیده میشود. برای این منظور، از کلمه ی **DECLARE** با فرمول زیر استفاده کنید:

**DECLARE @VariableName DataType**

کلمه ی **DECLARE** به **interpreter** اطلاع میدهد که شما **variable** را اعلام کرده اید. در **Transact-SQL** اسم متغیر با علامت **@** شروع میشود. اسم متغیر به شما اجازه میدهد که حافظه مورد نیاز برای ذخیره آن را مشخص کنید. در **Transact-SQL** مورد اسمی انعطاف زیادی دارد. برای مثال یک اسم میتواند فقط از اعداد تشکیل شده باشد. مثال:

**DECLARE @264**

بهتر است قوانین زیر را هنگام استفاده از اسمی رعایت کنید:

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

يك سام ميتواند با \_ يا حرف شروع شود مثل : @\_n, @act, @Second

بعد از حرف اول ميتوان يك عدد يا حرف يا \_ قرار داد مثل : @\_n24 or @act\_52\_t

قرار دادن space مجاز نيست

اسم نمیتواند شامل کاراکترهاي خاص باشد مثل : !, @, #, \$, %, ^, &, \*

اگر اسم شامل چند کلمه باشد حرف اول هر اسم بايد حرف بزرگ باشد: @DateHired, @\_RealSport,

@DriversLicenseNumber

اسم نبايد reserved word باشد.

هنگام اعلام کردن متغير , بعد از مشخص کردن اسم بايد **datatype** را هم مشخص کنید.

میتوان چند متغير را همزمان **declare** کرد. براي اين کار آنها را به وسيله ي ويرگول از هم جدا کنید. از فرمول زیر استفاده کنید :

**DECLARE @Variable1 DataType1, @Variable2 DataType2, @Variable\_n DataType\_n;**

بر خلاف زبانهاي ديگر , اگر متغيرهاي مختلف با **datatype** مشترك را **declare** ميکنيد , بعد از هر متغير بايد **datatype** آن را بنويسيد.

مقدار دهی اولیه متغير (initializing)

بعد از **declare** کردن متغير , مفسر فضايي را براي ذخيره به آن اختصاص ميدهد ولي اين به معنای پرکردن آنی فضا نيست. اين به اين معناست که که متغير ابتدا خالي يا **null** است. براي عوض کردن اين وضعيت کافي است به آن يك مقدار بدهيد. اين کار را مقدار دهی اولیه گویند.

براي يان کار, در جاي لازم , کلمه ي **SELECT** يا **SET** را تايپ کرده و سپس اسم متغير و بعد عمل = و در نهايت مقدار را تايپ کنید . از فرمول زیر استفاده کنید :

**SELECT @VariableName = DesiredValue**

يا

**SET @VariableName = DesiredValue**

آدرس آموزشگاه : تهران - خيابان شريعتی - بالا تر از خيابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

زمانی که مقداردهی متغیر انجام شد، می‌توانید آن را در دست‌نویس گذاشته یا نمایش دهید. می‌توانید اسم متغیر را سمت راست **PRINT** یا **SELECT** تایپ کنید.

استفاده از متغیر

مقدمه

بعد از تعیین اسم برای متغیر، باید میزان حافظه‌ی مورد احتیاج آن را مشخص کنید. چون **Database** می‌تواند با انواع مختلفی از اطلاعات کار کند، **SQL** مجموعه‌ای از **datatype** ها را تعبیه کرده است. نوع مورد استفاده برای متغیر دقیقاً مانند نوع مورد استفاده برای ستون‌هاست. این همچنین به این معناست که قوانین مورد استفاده برای **datatype** ها هم یکسان هستند که آنها را برای یادآوری به اختصار بررسی خواهیم کرد.

متغیرهای **boolean**

متغیر بولی به وسیله‌ی **datatype**، **BIT** یا **bit** تعریف میشوند. مثال:

```
DECLARE @IsOrganDonor bit;
```

بعد از مشخص کردن متغیر بولی، می‌توانید آن را با **0** یا هر عدد دیگری مقدار دهی کنید. **0** برای مقادیر غلط و باقی اعداد برای عبارات درست به کار می‌روند. مثال:

تمرین: استفاده از متغیرهای بولی

1. در **query editor** عبارت زیر را تایپ کنید:

2. برای اجرا دکمه‌ی **F5** را فشار دهید

اعداد صحیح

**Transact-SQL** از انواع مختلف اعداد پشتیبانی میکند. اگر متغیر شامل عددی طبیعی بین **-2,147,483,648** تا **2,147,483,647** باشد می‌توانید از **datatype**، **int** برای تعریف آن استفاده کنید.

اگر متغیر عددی مثبت و بین **0** تا **255** بود از **datatype** **tinyint** استفاده کنید

قوانین مورد استفاده برای **smallint** همانند **int** است به جز اینکه **smallint** برای ذخیره سازی اعداد کوچکتر بین **-32,768** و **32,767** به کار می‌رود.

آدرس آموزشگاه: تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

bigint Datatype برای متغیرهای کوچک یا خیلی بزرگ بین 9,223,372,036,854,775,808 تا استفاده میشود

تمرین : استفاده از اعداد صحیح به عنوان متغیر

1. جمله را به صورت زیر تغییر دهید :

2. عبارت را اجرا کنید

اعداد decimal

Transact-SQL از اعداد ده دهی پشتیبانی میکند. به طور مثال, میتوانید از numeric datatype یا decimal برای متغیرهایی که شامل همه ی اعداد هستند (اعداد طبیعی یا ده دهی) استفاده کنید.

به تعداد ارقام به کار رفته برای یک عدد دقت میگویند. همانطور که قبلاً مشاهده شد, برای مشخص کردن دقت اعداد از نوع decimal یا numeric یک پرانتز به datatype اضافه کنید و در داخل آن یک عدد بین 1 تا 38 بگذارید.

این عدد تعداد ارقام اعشار در سمت راست ممیز را تعیین میکند.

تمرین : استفاده از متغیرهای ده دهی

1. جمله را به صورت زیر تغییر دهید

2. عبارت را اجرا کنید

اعداد حقیقی

Transact-SQL به وسیله ی float datatype و real از اعداد اعشاری پشتیبانی میکند. مثال زیر استفاده از float datatype را برای نشان دادن اعداد اعشاری را نشان میدهد:

مقادیر پولی

اگر از متغیرهایی استفاده میکنید که شامل مقادیر مالی هستند, میتوانید از money datatype برای نشان دادن آنها استفاده کنید.

استفاده از datatype smallmoney از 214,748.3647 تا 214,748.3648 به خاطر داشته باشید که میتوانید برای مقادیر بین 4 قبل روی عدد ثابت شده است smallmoney و money کنید. دقت و مقیاس

تمرین : استفاده از متغیرهای مالی

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

1. عبارت را به صورت زیر تغییر دهید.

2. عبارت را اجرا کنید.

## Characters

برای نمایش متغیرهایی که از کاراکتر یا هر سمبل دیگری استفاده میکنند، میتوانید از **datatype char** استفاده کنید. برای مشخص کردن مقدار متغیر آن را در ( ) قرار دهید.

اگر متغیر شامل کاراکترهای بین المللی یا غیر لاتین بود (**unicode**) از **nchar datatype** استفاده کنید. هنگام تعیین مقدار متغیر، بهتر است قبل از آن (**N**) قرار دهید.

## رشته ها ( strings )

یک رشته مجموعه ای از کاراکتر یا نماد های مختلف است. برای مشخص کردن متغیرهایی از این دست **varchar datatype** را به کار ببرید.

بهتر است همیشه به یاد داشته باشید که طول رشته را باید مشخص کنید. برای این منظور یک عدد را در پرانتز جلوی آن قرار دهید. همچنین برای تعیین مقدار یک متغیر آن را در ( ' ) قرار دهید.

اگر از **Command Prompt** استفاده میکنید مقدار بین ( " ) قرار دهید.

اگر از **Query Editor** استفاده میکنید لازم نیست مقدار را در ( " ) قرار دهید. در غیر این صورت پیام خطا دریافت خواهید کرد.

موقع مقدار دهی استفاده کنید **nvarchar datatype** بود از **unicode** یا نمادهای **international** اگر متغیر شامل کاراکترهای قرار دهید **N** قبل از آن

استفاده کنید و این کار پیام خطایی به همراه نخواهد داشت **N** از **nvarchar** یا **varchar** , **nchar** , **char** میتوانید برای مقدار دهی

فقط کاراکتری که در جای اول از سمت چپ قرار گرفته در متغیر ذخیره خواهد، چنانچه بیشتر از یک عبارت را در ( ' ) میگذارید شد.

تمرین : استفاده از متغیرهای رشته ای

1. عبارت را به صورت زیر تغییر دهید.

2. عبارت را اجرا کنید.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

3. file را به اسم variable در mydocument ذخیره کنید.

متن

اگر متغیر از يك متن بزرگ استفاده میکند , `varchar(max) datatype` را به کار ببرید.:

اگر متن شامل کاراکترهاي `unicode` باشد `nvarchar(max) datatype` را به کار ببرید.

برای مقدار دهی متغیر از قوانینی که تا اینجا بحث کردیم استفاده کنید.

### SQL variants

**Transact-SQL** برای متغیرهایی که میتوانند همه نوع مقادیر را نگه دارند , `sql_variant datatype` را تعبیه کرده است. هنگام مقدار دهی به متغیر باید از قوانین `sql variant` پیروی کنید.

عبارات هندسی

برای پشتیبانی از مختصات هندسی از `geometry datatype` استفاده کنید.

**Geometry** شامل ویژگی و روش هایی است. برای مشخص کردن متغیر , `geometry` ابتدا باید به آن یک مقدار دهید. این کار به وسیله ی روش `STGeomFromText` انجام پذیر است که دستور آن به شرح زیر است :

این روش ایستا است. این به این معناست که برای دسترسی به آن شما از `geometry::STGeomFromText` استفاده میکنید.

این روش شامل دو بخش است. بخش اول نگهداری مقدار به صورت **Well-Known Text (WKT)** است. مقدار از فرمت تعریف شده ی **OGC** تبعیت میکند. راه های مختلفی برای تعیین مقدار وجود دارد. همانطور که قبلاً مشاهده کردید , یک مختصات هندسی شامل دو مقدار است. یکی مختصات افقی یا `x` و دیگری مختصات عمودی یا `y`. این دو مقدار میتوانند یک عدد صحیح یا اعشاری باشند.

چنانچه مختصات یک نقطه را دارید و میخواهید از آن به صورت `geometry` استفاده کنید , کلمه ی `point()` را تایپ کرده و در پرانتز مختصات را تایپ کرده و به وسیله ی `space` از هم جدا کنید.

همچنین میتوانید به جای یک نقطه میتوانید مختصات یک خط را وارد نمایید. برای این منظور میتوانید از `linestring()` استفاده کنید. کافی است `x` و `y` را در دو طرف ویرگول تایپ کنید.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

اگر مایل به استفاده از مقادیر مرکب هستید می‌توانید از شناسه **polygon** استفاده کنید. ابتدا **polygon()** را تایپ کنید سپس مختصات هر نقطه را داخل پرانتز تایپ کنید. و به وسیله **Y** ویرگول از هم جدا کنید. آخرین نقطه که برای بستن پرانتز از آن استفاده میشود باید با اولین نقطه برابر باشد. مثال :

دومین روش از **geometry::STGeomFromText** عدد ثابت یا **spatial reference ID (SRID)** است. بعد از نمایش و مقدار دهی , می‌توانید از عبارت **select** برای نمایش مقدار استفاده کنید.

نمایش محل جغرافیایی

**Transact-SQL** از محل های جغرافیایی پشتیبانی میکند.

عبارات از پیش تعیین شده

می‌توانید یک نوع از **datatype** های موجود را از پیش تعیین کنید. به این کار **user-defined data type (UDT)** می‌گویند. بیشتر چگونگی ایجاد آنها را بررسی کردیم.

بعد از تشکیل **UDT** , می‌توانید یک متغیر برای آن مشخص کنید. سپس قبل از استفاده باید ابتدا به آن مقدار بدهید.

البته می‌توانید **Transact-SQL data types** را با **datatype** های خودتان در کد میکس کنید.

عملیتهای مرکب

مقدمه

عملیات مرکب عبارات است از انجام یک عملیات ریاضی بر روی خود متغیر. به طور مثال , فرض کنید که یک متغیر دارید و میخواهید روی آن عملیاتی انجام داده و مقدار متغیر را با خودش جمع کنید. عملیات مرکب از عملگری استفاده میکند که در واقع مجموعه از دو عملگر است. متغیر میتواند از هر نوعی باشد به شرطی که عملیات بر روی آن قابل انجام باشد.

مثال : عملیات مرکب

عملیات مرکب از عمل کننده **+=** استفاده میکند. به وسیله **Y** آن میتوان یک مقدار را به خودش اضافه کرد. برای این کار ابتدا متغیر و خودش را تایپ کرده و عملگر را بین آنها قرار دهید. بعد از انجام این عملیات , متغیر مقدار جدیدی خواهد داشت. شکل زیر را در نظر بگیرید :

متغیر به کار رفته در عملیات مرکب میتواند از هر نوعی باشد به شرط آن نوع از عملیات پشتیبانی کند. به عنوان مثال رشته ها از عملیات جمع پشتیبانی می کنند. این به این معناست که متغیر میتواند از نوع **char** و مشتقات آن باشد.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

همچنین میتوان یک مقدار را به متغیر اضافه کرد. برای این کار ابتدا متغیر و مقداری که میخواهید به آن اضافه کنید را تایپ کرده و عملگر را بین آنها قرار دهید.

زمانی که عملیات انجام شد، متغیر سمت چپ دارای مقداری جدید خواهد بود. :

عملیات مرکب اضافه کردن یک مقدار ثابت به متغیر است. برای این کار عملگر += را قرار داده و مقدار یکی دیگر از کاربردهای ثابت را در سمت راست آن بگذارید

به یاد داشته باشید که بعد از انجام عملیات متغیر دارای مقداری جدید خواهد بود.

همچنین میتوانید این عملیات را چندین بار تکرار کنید

همیشه ظرفیت ذخیره عملگر سمت چپ را به یاد داشته باشید. عملگر باید قابلیت همه ی مقادیری که به آن اضافه میشود را داشته باشد.

### دیگر عملیتهای دونایی

عملیات مرکب برای در و اقع شامل 2 عملیات ریاضی است. همانطور که در بالا مشاهده شد، عملیات مرکب برای رشته ها نیز امکان پذیر است. همچنین برای دستکاری بیت ها قابل استفاده است. به یاد داشته باشید که همه ی datatype ها قابلیت استفاده از عملیات مرکب را ندارند.

### جمع بندی :

همه ی اعداد دودویی از جمع، تفریق، ضرب، تقسیم و باقیمانده پشتیبانی میکنند. عملگرها به ترتیب /, \*, -, += و & هستند.

عملیات تنظیم بیت برای موارد زیر قابل استفاده است :

o استفاده از عملگر &= برای مقایسه بیت سمت چپ و سمت راست عملگر و بررسی مساوی بودن آنها

o استفاده از عملگر or یا |= برای مقایسه بیت سمت چپ و سمت راست عملگر و بررسی نامساوی بودن آنها

o استفاده از عملگر xor یا ^=

این عملگرها برای اعداد دهدهی و طبیعی نیز قابل استفاده هستند.

تمرین : ویرایش درس

1. برنامه را ببندید

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330



2. در صورت درخواست ذخیره روی **no** کلیک کنید.



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>