

(T2)入門 DataTyp、DDL、DML

CourseGUID: e48417fc-9db5-4e99-822c-706c5ccef6cc

---

(T2)入門 DataTyp、DDL、DML

---

## 1. Data Type

1.2. Exact numerics : decimal and numeric

1.3. Exact numerics : money and smallmoney

1.4. Approximate numerics : float and real

1.5. Date and Time

1.6. Char, varchar, varchar(Max), nchar, nvarchar, nvarchar(Max)

1.7. uniqueidentifier

## 2. DDL V.S. DML / Fragmentation and Defragmentation

---

# 1. Data Type

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql>

[https://technet.microsoft.com/en-us/library/ms187752\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187752(v=sql.105).aspx)

## 1.1. Exact numerics : int, bigint, smallint, and tinyint

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/int-bigint-smallint-and-tinyint-transact-sql>

### **bigint**

-2<sup>63</sup> (-9,223,372,036,854,775,808) to 2<sup>63</sup>-1 (9,223,372,036,854,775,807)

8 Bytes

### **int**

-2<sup>31</sup> (-2,147,483,648) to 2<sup>31</sup>-1 (2,147,483,647)

4 Bytes

### **smallint**

-2<sup>15</sup> (-32,768) to 2<sup>15</sup>-1 (32,767)

2 Bytes

### **tinyint**

0 to 255

1 Byte

## 1.2. Exact numerics : decimal and numeric

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/decimal-and-numeric-transact-sql>

**decimal** [ (p [ ,s ] ) ] and **numeric** [ (p [ ,s ] ) ]

The numeric and decimal types vary

because you can specify how large of a decimal portion you want to be able to store,

and so the more accuracy or the more numbers you keep, the larger the data takes on disk.

Precision	Storage bytes
1 - 9	5
10-19	9
20-28	13
29-38	17

```
CREATE TABLE dbo.MyTable (
    MyDecimalColumn decimal(5,2) ,
    MyNumericColumn numeric(10,5)
);
```

--12.345 is converted into a **numeric** value with a **precision** of **5** and a scale of **3**.

## 1.3. Exact numerics : money and smallmoney

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/money-and-smallmoney-transact-sql>

Data types that represent monetary or currency values.

### money

-922,337,203,685,477.5808 to 922,337,203,685,477.5807 (-922,337,203,685,477.58 to 922,337,203,685,477.58 for Informatica. Informatica only supports two decimals, not four.)  
8 bytes

### smallmoney

- 214,748.3648 to 214,748.3647  
4 bytes

## 1.4. Approximate numerics : float and real

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/float-and-real-transact-sql>

**float [ (n) ]** Where n is the number of bits that are used to store the mantissa of the float number in scientific notation and, therefore, dictates the precision and storage size. If n is specified, it must be a value between **1 and 53**. The default value of n is **53**.

The ISO synonym for **real** is **float(24)**.

n value	Precision	Storage size
1-24	7 digits	4 bytes
25-53	15 digits	8 bytes

### float

- 1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308  
Depends on the value of n

### real

- 3.40E + 38 to -1.18E - 38, 0 and 1.18E - 38 to 3.40E + 38  
4 Bytes

## 1.5. Date and Time

Data type	Format	Range	Accuracy	Storage size (bytes)
time	hh:mm:ss[.nnnnnnn]	00:00:00.0000000 through 23:59:59.9999999	100 nanoseconds	3 to 5
date	YYYY-MM-DD	0001-01-01 through 9999-12-31	1 day	3
smalldatetime	YYYY-MM-DD hh:mm:ss	1900-01-01 through 2079-06-06	1 minute	4
datetime	YYYY-MM-DD hh:mm:ss[.nnn]	1753-01-01 through 9999-12-31	0.00333 second	8
datetime2	YYYY-MM-DD hh:mm:ss[.nnnnnnn]	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999	100 nanoseconds	6 to 8
datetimeoffset	YYYY-MM-DD hh:mm:ss[.nnnnnnn] [+ -]hh:mm	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999 (in UTC)	100 nanoseconds	8 to 10

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql>

## 1.6. Char, varchar, varchar(Max), nchar, nvarchar, nvarchar(Max)

### Char(N),

is a fixed length characters.

Each **character** take **1 Byte**.

It is like C# string with the length of N.

If the value is "KL", and if the data type is Char(5).

Then it will fill 3 blank into the string, `___KL`

**varchar(N)** is not a fixed length characters.

Each **character** take **1 Byte**.

If the value is "KL", and if the data type is Char(5).

Then it will not fill blanks into the string, "KL"

### varchar(Max)

It is similar to **varchar(N)**

Just replace N to max value.

Each **character** take **1 Byte**.

### NChar(N),

It is similar to **Char(N)**

But each **character** take **2 Byte** in order to save **non-English language character** such as **Chinese character**

### Nvarchar(N)

It is similar to **varchar(N)**

But each **character** take **2 Byte** in order to save **non-English language character** such as **Chinese character**

### Nvarchar(Max)

It is similar to **varchar(Max)**

But each **character** take 2 **Byte** in order to save **non-English language character** such as **Chinese character**

## 1.7. uniqueidentifier

Reference:

<https://docs.microsoft.com/en-us/sql/t-sql/data-types/uniqueidentifier-transact-sql>

### uniqueidentifier

16 bytes.

E.g.

6F9619FF-8B86-D011-B42D-00C04FC964FF

## 2. DDL V.S. DML / Fragmentation and Defragmentation

<https://stackoverflow.com/questions/2578194/what-is-ddl-and-dml>

[https://technet.microsoft.com/en-us/library/ff848799\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/ff848799(v=sql.110).aspx)

### Data Definition Language (DDL)

- It is used to define data **structures**.
- Data Definition Language (DDL) statements are used to define the database structure or **schema**. Some examples:
- SQL Server uses **Transact-SQL**, or **T-SQL**

1. CREATE - to create objects in the database
2. ALTER - alters the structure of the database
3. DROP - delete objects from the database
4. **TRUNCATE** 截短 - removes rows from a table and reclaims free space. The data is **less likely** to become **fragmented** and more likely to stay in the correct order.
5. COMMENT - add comments to the data dictionary
6. RENAME - rename an object

### Data Manipulation Language (DML)

- It is used to manipulate **data itself**.
- Data Manipulation Language (DML) statements are used for managing data within schema objects. Some examples:

1. SELECT - retrieve data from the a database
2. INSERT - insert data into a table UPDATE - updates existing data within a table
3. **DELETE** - removes rows from a table but does not reclaim space. It does so piecemeal 零碎地, which can result in something called **fragmentation**.
4. MERGE - UPSERT operation (insert or update)
5. CALL - call a PL/SQL or Java subprogram
6. EXPLAIN PLAN - explain access path to data
7. LOCK TABLE - control concurrency

### Fragmentation and Defragmentation

Over time, fragmentation can reduce SQL Server's performance,

and you have to do something called defragmentation to put the pages back in the order that they're meant to be in.