Типы данных

Числовые типы

Точные числовые типы

К категории точных числовых типов в SQL относятся те типы, значения которых точно представляют числа. Типы данных этой категории распадаются на две части: целые типы (INTEGER и SMALLINT) и типы, допускающие наличие дробной части (NUMERIC и DECIMAL).

- **целочисленные**:
  tinyint 0-255,
  smallint *(от -32 768 до 32 767),*
  int *(от -2,147,483,648 до 2,147,483,647) и*
  bigint (от -2^63 до 2^63);

- **десятичные**: **decimal** и **numeric** (это - два названия одного и того же типа);

- **денежные**: **money** (от -2^63 до 2^63 - с точностью 4 знака после запятой) и **smallmoney** (от -214748.3648 до +214748.3647).;
- **с плавающей запятой**: **float** (от -1.79E + 308 до 1.79E + 308) и **real** (от -3.40E + 38 до 3.40E + 38).

с плавающей запятой:

float (от -1.79E + 308 до 1.79E + 308) и **real** (от -3.40E + 38 до 3.40E + 38

DECIMAL [(точность[,масштаб])]

Параметр **точность** указывает максимальное количество цифр вводимых данных этого типа (до и после десятичной точки в сумме), а параметр **масштаб** – максимальное количество цифр, расположенных после десятичной точки.

Строковые типы

В SQL Server предусмотрены две дублирующих разновидности полей для представления текстовых данных:

поля **Unicode** и не-**Unicode**.

Unicode - типы данных начинаются символом **n** (от слова national, то есть с поддержкой национальных символов).

Всего в SQL Server предусмотрены следующие типы для текстовых данных:

- **char/nchar** - строковые данные фиксированной длины;
- **varchar/nvarchar** - строковые данные переменной длины.

- При использовании типа **Char** значения длиной короче заданной дополняются пробелами до указанной длины. Максимальное значение длины – 8000 символов.
- При использовании типа **VarChar** значения длиной короче заданной не дополняются пробелами.

Если необходимо ввести значения большой длины можно использовать ключевое слово **max**, что позволяет определять столбцы до 2^{31} байтов.

varchar(max).

- **datetime** (8 байт, точность до 3,33 миллисекунд);
- **smalldatetime** (4 байта, точность до минуты).

В большинстве приложений вполне хватает **smalldatetime;**

Тип данных **UNIQUEIDENTIFIER** используется для хранения глобальных уникальных идентификационных номеров.

SQL_VARIANT -
Служит для хранения значений разных типов одновременно, таких как числовые значения, строки и даты.

Объявлять тип столбца как SQL_VARIANT следует только в том случае, если это действительно необходимо. Например, если столбец предназначается для хранения значений разных типов данных или если при создании таблицы тип данных, которые будут храниться в данном столбце, неизвестен.

Логический тип данных - хранит значения вида **true/false** (единица/ноль).

В SQL Server он представлен типом данных **boolean**.

Функции даты

- **GETDATE** ( ) –

Возвращает значение типа datetime, которое содержит дату и время компьютера, на котором запущен экземпляр SQL Server.

- **DATEDIFF** (datepart , startdate , enddate)– возвращает интервал времени, прошедшего между двумя временными отметками - *startdate* (начальная отметка) и *enddate* (конечная отметка). Этот интервал может быть измерен в разных единицах. Возможные варианты определяются аргументом *datepart*

DATEPART (*datepart* , *date*) — возвращает целое число, представляющее собой указанную аргументом *datepart* часть заданной вторым аргументом даты

Year - год	yy, yyyy
Quarter - квартал	qq, q
Month - месяц	mm, m
Dayofyear - день года	dy, y
Day - день	dd, d
Week - неделя	wk, ww

В ряде случаев функцию **DATEPART** можно заменить более простыми функциями.

DAY (*date*) - целочисленное представление дня указанной даты. Эта функция эквивалентна функции **DATEPART**(*dd, date*).

MONTH (*date*) - целочисленное представление месяца указанной даты. Эта функция эквивалентна функции **DATEPART**(*mm, date*).

YEAR (*date*) - целочисленное представление года указанной даты. Эта функция эквивалентна функции **DATEPART**(*yy, date*).

Примеры:

DATEPART(dd, '01.01.2016’) – возвращает день месяца (число
для указанной даты

такой же результат будет получен с помощью функции **DAY**
DAY('01.01.2016’)

Найти разность дат в днях между датой отъезда ***time_out*** ***и датой приезда time_in***

DATEDIFF(dd, time_out, time_in)

Возвращает новое значение **datetime**, добавляя интервал к указанной части *datepart* заданной даты *date*.

DATEADD (*datepart, number , date*)

Пользовательские типы данных.

Могут использоваться при определении какого-либо специфического или часто употребляемого формата.

Создание пользовательского типа данных осуществляется выполнением системной процедуры:

sp_addtype [@typename=]type,[@phystype=] system_data_type [,[@nulltype=]'null_type']

EXEC sp_addtype dt, DATETIME, 'NULL'

Удаление пользовательского типа данных происходит в результате выполнения процедуры sp_droptype type

Пример:

EXEC sp_droptype 'dt‘

http://www.intuit.ru/studies/courses/5/5/lecture/124?page=2

CREATE TYPE SSN
FROM varchar(10) NOT NULL ;

Преобразование типов

Для выполнения преобразований SQL Server содержит функции **CONVERT** и **CAST**, с помощью которых значения одного типа преобразовываются в значения другого типа, если такие изменения вообще возможны.

CONVERT и **CAST** могут быть взаимозаменяемыми.

CAST(выражение AS тип_данных)

CONVERT(тип_данных[(длина)], выражение)

Пример:

SELECT ‘сегодня ‘ + CONVERT(VARCHAR(11),GETDATE())

CAST('1977.01.07‘ AS Datetime)

Оновные функции

– поиск подстроки

CHARINDEX (expressionToFind ,expressionToSearch[, start_location])

Пример. SELECT CHARINDEX ('морф', 'полиморфизм') – Возвращает 5

- вырезка

SUBSTRING (**expression ,start , length**)

- REPLACE

заменяет указанную подстроку первого операнда строкой, заданной в качестве второго операнда.

REPLACE(expression , string_pattern , string_replacement)

-REVERSE

- Возвращает строковое значение, где символы переставлены в обратном порядке справа налево.

- TRIM "отсекает" последовательности указанного символа в конце или

Временные таблицы

Временные таблицы похожи на обычные, однако они не предназначены для постоянного хранения данных. Они создаются, удаляются и используются как обычные таблицы.

Имена временных таблиц должны начинаться с символов **#** или **##**.

Временные таблицы удаляются при отключении пользователя от базы данных.

Временные таблицы используются так, как будто они входят в текущую базу данных, однако в действительности данные хранятся в **TEMPDB.**

В SQL Server существуют два типа временных таблиц: **локальные** и **глобальные**.

Локальные временные таблицы доступны лишь для своего владельца. Имена локальных временных таблиц начинаются с префикса #.

Глобальные временные таблицы доступны для всех пользователей, их имена должны начинаться с префикса **##**.