

### Лабораторная работа №3.1 Язык SQL

**Основные понятия языка Transact-SQL. Основные понятия языка T-SQL. Соглашения о синтаксисе T-SQL. Многокомпонентные имена. Идентификатор. Константы. Типы данных.**

#### 3.1.1 Теоретические сведения.

Transact-SQL структурированный язык запросов, являющийся инструментальным средством для описания, управления и манипулирования реляционными данными в СУБД Microsoft SQL Server.

**Идентификатор.** Любой объект в базе данных должен быть идентифицирован. Для этого ему присваивают имя

**Правила задания идентификатора:**

- Длина идентификатора от 1 до 128 символов.
- Идентификатор может включать любые символы, определенные стандартом Unicode Standard 2.0 кроме запрещенных символов.
- В идентификатор не должны входить слова, являющиеся зарезервированными.
- Первым символом в идентификаторе может быть: буква алфавита или символ подчеркивания «\_».
- Для обозначения **временных объектов (переменных и параметров)** разрешается использовать в качестве первого символа «@».
- Transact-SQL не различает регистр. Например, идентификатор *Фамилия* соответствует идентификатору *фамилия*.

**Типы идентификаторов:**

- @ - идентификатор локальной переменной (пользовательской).
- @@ - идентификатор глобальной переменной (встроенной).
- # - идентификатор локальной таблицы или процедур
- ## - идентификатор глобальной таблицы или процедуры.
- [ ] - идентификатор группировки слов в переменную.

Идентификатор, заданный в соответствии с этими правилами называется **обычным идентификатором**. Кроме обычных идентификаторов можно задавать идентификаторы с разделителем. **Идентификаторы с разделителем** заключаются в квадратные скобки ([ ]) или в двойные кавычки (" ") и может при необходимости содержать недопустимый символ или зарезервированные слова. Примеры идентификатора с разделителем: [Общая ведомость], [Номер группы], [Код специальности].

**Именованное.** SQL Server достаточно лояльно относиться к именам таблиц.

**Можно использовать в качестве имен русские слова, и даже имена из нескольких слов. Только если имя состоит из нескольких слов, оно должно заключаться в квадратные скобки.**

#### 3.1.2 Типы данных в Transact-SQL.

**Тип данных определяет диапазон значений, которые можно сохранить в переменной или в поле таблицы.**

**Числовые типы:**

**Числовые типы данных**

1) *Целочисленные типы данных* (общее название integer):

- **SMALLINT:** хранит числа от -32 768 до 32 767. Занимает 2 байта

- **INT:** хранит числа от -2 147 483 648 до 2 147 483 647. Занимает 4 байта. Наиболее используемый тип для хранения чисел.

- **BIT**: хранит значение от 0 до 16. Может выступать аналогом булевого типа в языках программирования (в этом случае значению true соответствует 1, а значению false - 0). При значениях до 8 (включительно) занимает 1 байт, при значениях от 9 до 16 - 2 байта.
- **TINYINT**: хранит числа от 0 до 255. Занимает 1 байт. Хорошо подходит для хранения небольших чисел.
- **BIGINT**: хранит очень большие числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, которые занимают в памяти 8 байт.

## 2) Нецелочисленные типы данных

Подразделяются на два типа: **десятичные** (decimal) и **приблизительные** (approximate).

Десятичные данные хранятся в виде последовательности цифр.

- **DECIMAL**: хранит числа с фиксированной точностью. Занимает от 5 до 17 байт в зависимости от количества чисел после запятой.

Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale).

Параметр precision представляет максимальное количество цифр, которые может хранить число. Это значение должно находиться в диапазоне от 1 до 38. По умолчанию оно равно 18.

Параметр scale представляет максимальное количество цифр, которые может содержать число после запятой. Это значение должно находиться в диапазоне от 0 до значения параметра precision. По умолчанию оно равно 0.

- **NUMERIC**: данный тип аналогичен типу DECIMAL.

Приблизительные типы используются для работы с данными, имеющими значения от очень малых величин до предельно больших. К ним относятся:

- **FLOAT**: хранит числа от  $-1.79E+308$  до  $1.79E+308$ . Занимает от 4 до 8 байт в зависимости от дробной части.

Может иметь форму определения в виде FLOAT(n), где n представляет число бит, которые используются для хранения десятичной части числа (мантиссы). По умолчанию n = 53.

- **REAL**: хранит числа от  $-340E+38$  to  $3.40E+38$ . Занимает 4 байта. Эквивалентен типу FLOAT(24).

## Денежные типы данных

Используются для хранения данных о денежных суммах. Позволяет хранить после запятой четыре знака. **К денежным типам относятся:**

- **SMALLMONEY**: хранит дробные значения от -214 748.3648 до 214 748.3647. Предназначено для хранения денежных величин. Занимает 4 байта. Эквивалентен типу DECIMAL(10,4).
- **MONEY**: хранит дробные значения от -922 337 203 685 477.5808 до 922 337 203 685 477.5807. Представляет денежные величины и занимает 8 байт. Эквивалентен типу DECIMAL(19,4).

## Тип даты и времени

Существует встроенные типы, позволяющие хранить сведения о дате и о времени одновременно, или отдельно. С данными этого типа можно выполнять арифметические операции сложения и вычитания, а также сравнивать их. К временным типам относятся:

- **DATE**: хранит даты от 0001-01-01 (1 января 0001 года) до 9999-12-31 (31 декабря 9999 года). Занимает 3 байта.
- **TIME**: хранит время в диапазоне от 00:00:00.0000000 до 23:59:59.9999999. Занимает от 3 до 5 байт.

Может иметь форму TIME(n), где n представляет количество цифр от 0 до 7 в дробной части секунд.

- **DATETIME**: хранит даты и время от 01/01/1753 до 31/12/9999. Занимает 8 байт.
- **DATETIME2**: хранит даты и время в диапазоне от 01/01/0001 00:00:00.0000000 до 31/12/9999 23:59:59.9999999. Занимает от 6 до 8 байт в зависимости от точности времени.

Может иметь форму DATETIME2(n), где n представляет количество цифр от 0 до 7 в дробной части секунд.

- **SMALLDATETIME**: хранит даты и время в диапазоне от 01/01/1900 до 06/06/2079, то есть ближайшие даты. Занимает от 4 байта.
- **DATETIMEOFFSET**: хранит даты и время в диапазоне от 0001-01-01 до 9999-12-31. Сохраняет детальную информацию о времени с точностью до 100 наносекунд. Занимает 10 байт.

## Строковые типы данных

- **CHAR**: хранит строку длиной от 1 до 8 000 символов. На каждый символ выделяет по 1 байту. Не подходит для многих языков, так как хранит символы не в кодировке Unicode. Используется для хранения набора символов длиной n ( $n_{\max} = 8000$ ). Строка данного типа является строкой фиксированной длины. Если строка содержит символов меньше, чем n, то в конец строки добавляются пробелы. Если символов больше n, то часть конечных символов будет потеряна. Количество символов, которое может

хранить столбец, передается в скобках. Например, для столбца с типом CHAR(10) будет выделено 10 байт. И если мы сохраним в столбце строку менее 10 символов, то она будет дополнена пробелами.

- **VARCHAR**: хранит строку. На каждый символ выделяется 1 байт. Можно указать конкретную длину для столбца - от 1 до 8 000 символов, например, VARCHAR(10). Если строка должна иметь больше 8000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб: VARCHAR(MAX).

Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

В отличие от типа CHAR если в столбец с типом VARCHAR(10) будет сохранена строка в 5 символов, то в столбец будет сохранено именно пять символов.

- **NCHAR**: хранит строку в кодировке Unicode длиной от 1 до 4 000 символов. На каждый символ выделяется 2 байта. Например, NCHAR(15)

- **NVARCHAR**: хранит строку в кодировке Unicode. На каждый символ выделяется 2 байта. Можно задать конкретный размер от 1 до 4 000 символов: . Если строка должна иметь больше 4000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб.

**Примечание.** Для использования русских символов (не ASCII кодировки) используются типы данных с приставкой "n" (nchar, nvarchar, ntext), которые кодируют символы двумя байтами. Иначе говоря, для работы с Unicode используются типы данных с "n".

**Примечание.** Для данных переменной длины используются типы данных с приставкой "var". Типы данных без приставки "var" имеют фиксированную длину области памяти, неиспользованная часть которой заполняется пробелами или нулями.

### 3.1.3 Использование конструктора таблиц в SQL Server Management Studio

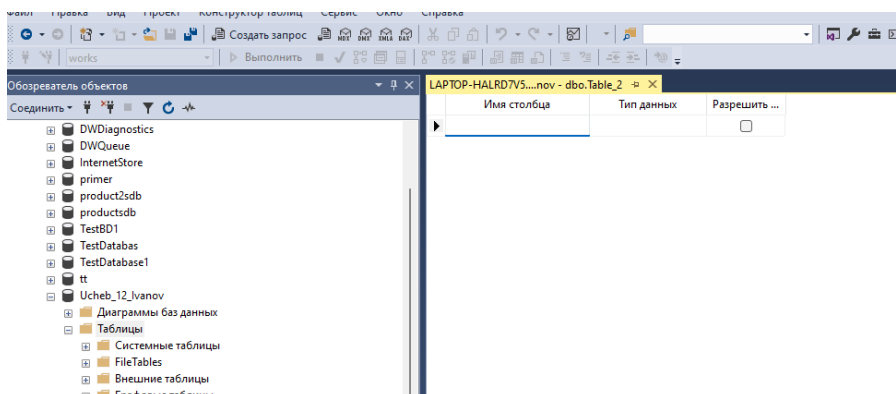
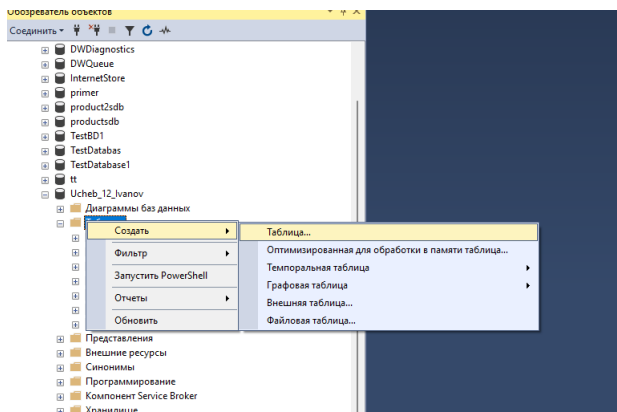
#### Задание 1

1) **Создайте БД согласно п.1.7 Лабораторная\_1\_часть 2 (работа в SSMS) (стр.16)**

2) **Создание таблиц средствами SSMS**

В обозревателе объектов *разверните узел Базы данных*, а затем базу данных, в которой будет размещена новая таблица.

В обозревателе объектов щелкните правой кнопкой мыши узел *Таблицы* базы данных и выберите *Создать таблицу*.



Вводим имена столбцов, выберите типы данных и определите для каждого столбца, могут ли в нем присутствовать значения NULL.

Например,

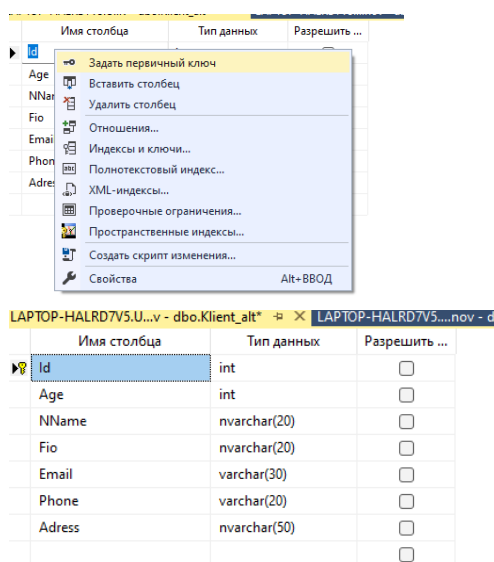
Имя столбца	Тип данных	Разрешить ...
Id	int	<input type="checkbox"/>
Age	int	<input type="checkbox"/>
NName	nvarchar(20)	<input type="checkbox"/>
Fio	nvarchar(20)	<input type="checkbox"/>
Email	varchar(30)	<input type="checkbox"/>
Phone	varchar(20)	<input type="checkbox"/>
Adress	nvarchar(50)	<input type="checkbox"/>

Чтобы указать дополнительные свойства столбца, например идентификаторы или вычисляемые значения столбца, **выберите столбец и на вкладке свойств столбца выберите соответствующие свойства. Примечание, описание свойств см. раздел 1.3**

Чтобы указать, что столбец **является столбцом первичного ключа**, щелкните его правой кнопкой мыши и выберите **Задать первичный ключ**.

**Первичный ключ** это одно или несколько полей в таблице. Он необходим для уникальной идентификации любой строки. Первичный ключ накладывает некоторые ограничения. Все записи относящиеся к первичному ключу должны быть уникальны. Записи в полях относящихся к первичному ключу не могут быть пустыми. В каждой таблице может присутствовать только один первичный ключ.

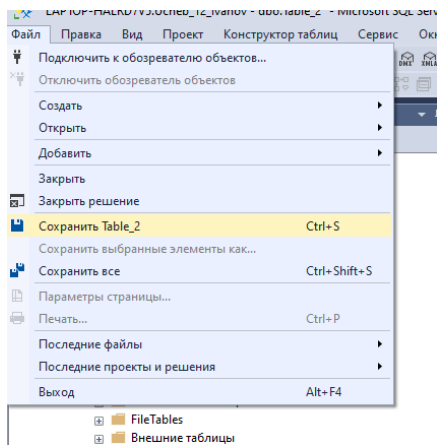
**«Первичный ключ — это значение или комбинация нескольких значений из таблицы, уникально определяющая каждую запись в этой таблице. стол. Если мы знаем это значение/комбинацию, мы можем легко найти связанную запись и получить доступ ко всем оставшимся значениям из этой записи».**



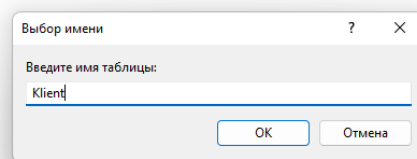
The screenshot shows the 'Columns' tab of a table in SQL Server Enterprise Manager. The 'Id' column is selected, and a right-click context menu is open with 'Set Primary Key' (Задать первичный ключ) highlighted. Below the menu, the table structure is visible with columns: Id (int), Age (int), NName (nvarchar(20)), Fio (nvarchar(20)), Email (varchar(30)), Phone (varchar(20)), and Adress (nvarchar(50)).

По умолчанию таблица содержится в схеме dbo. **Префикс dbo обозначает, что таблица является объектом БД (Data Base Object).** В дальнейшем при работе с объектами БД префикс dbo можно опускать.

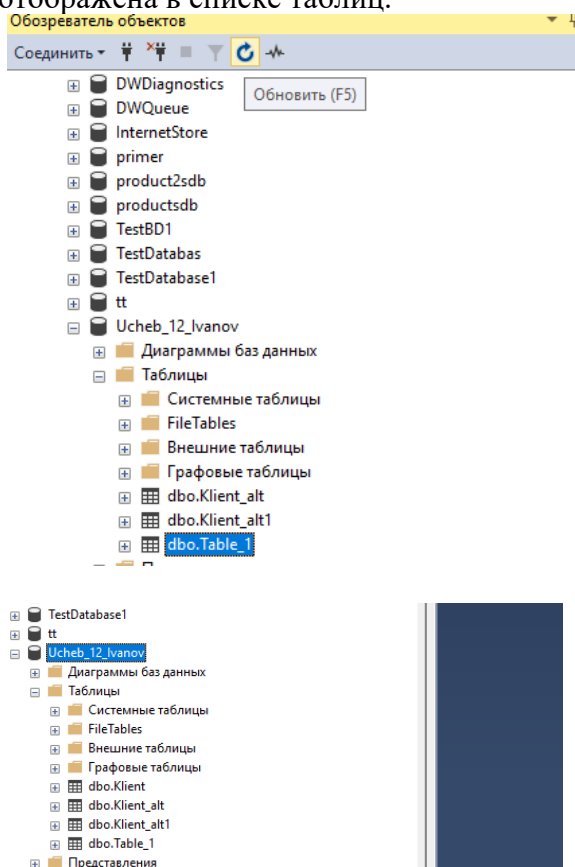
В меню Файл выберите команду **Сохранить имя\_таблицы**.



В диалоговом окне Выбор имени введите имя таблицы и нажмите кнопку ОК.



Чтобы просмотреть новую таблицу, в обозревателе объектов разверните узел Таблицы , а затем нажмите клавишу F5 , чтобы обновить список объектов. Новая таблица будет отображена в списке таблиц.



### **1.3 Свойства столбца таблицы (среда SQL Server Management Studio)**

Эти свойства отображаются на нижней панели конструктора таблиц. Если не оговорено обратное, эти свойства можно изменять в окне «Свойства» при выборе столбца.

### **Общие сведения**

**Имя.** Отображает имя выбранного столбца.

**Разрешить значения NULL.** Указывает, может ли столбец содержать значения NULL. Для изменения этого свойства щелкните флажок «Разрешить значения NULL», соответствующий столбцу на верхней панели конструктора таблиц.

**Тип данных.** Отображает тип данных выбранного столбца. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Значение по умолчанию или привязка.** Выводит значение по умолчанию для этого столбца, устанавливаемое, если никакого значения не было присвоено. Значение этого поля может быть либо значением ограничения SQL Server по умолчанию, либо именем глобального ограничения, к которому привязан столбец. Раскрывающийся список содержит все глобальные значения по умолчанию, определенные в базе данных. Для привязки столбца к глобальным значениям по умолчанию выберите значение из раскрывающегося списка. Или же, чтобы создать ограничение столбца по умолчанию, введите значение по умолчанию непосредственно в текстовом виде.

**Длина.** Указывает число символов, допустимых для символьных типов данных. Данное свойство доступно только для символьных типов данных/

**Масштаб.** Отображает максимальное количество цифр справа от десятичной запятой в значениях этого столбца. Для нечисловых типов данных в этом свойстве отображается 0.

**Точность.** Выводит максимальное число цифр в значениях этого столбца. Для нечисловых типов данных в этом свойстве отображается 0 .

### **Конструктор таблиц**

**Параметры сортировки.** Отображает последовательность сортировки, которая SQL Server применяется по умолчанию к столбцу всякий раз, когда значения столбцов используются для сортировки строк результата запроса. Чтобы изменить параметры сортировки, выберите свойство, щелкните многоточие (...), которое отображается справа от значения свойства, чтобы открыть диалоговое окно Параметры сортировки .

**Спецификация вычисляемого столбца.** Выводит данные о вычисляемом столбце. Значение этого свойства совпадает со значением свойства потомка Формула и выводит эту формулу для данного вычисляемого столбца.

Примечание: Для изменения значения свойства Спецификация вычисляемого столбца необходимо развернуть его и изменить свойство потомка Формула.

**Формула.** Выводит формулу для данного вычисляемого столбца. Чтобы редактировать это свойство, введите в поле новую формулу.

**Сохранен.** Показывает, сохранены ли результаты вычислений по этой формуле. Если это свойство равно Нет, то сохраняется только формула, а значения вычисляются при каждом обращении к этому столбцу. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Сжатый тип данных.** Отображает сведения о типе данных поля в том же формате, что и в инструкции SQL CREATE TABLE. Например: поле, содержащее строку переменной длины с максимальной длиной 20 символов, будет представлено как «varchar(20)». Чтобы изменить это свойство, введите значение непосредственно в поле.

**Описание.** Выводит текст с описанием этого столбца. Чтобы изменить описание, выберите свойство , щелкните многоточие (...), которое отображается справа от значения свойства, и измените описание в диалоговом окне Свойство описания .

**Детерминированное.** Указывает, может ли тип данных для выбранного столбца быть определен точно.

**Опубликован через службы DTS.** Указывает, опубликован ли столбец через службы DTS.

**Спецификация полного текста.** Выводит данные о полнотекстовом индексе. Значение этого свойства совпадает со значением свойства потомка С полнотекстовым индексом и указывает, построен ли для данного столбца полнотекстовый индекс.

**С полнотекстовым индексом.** Указывает, построен ли для данного столбца полнотекстовый индекс. Этому свойству может быть присвоено значение Да только в том случае, если тип данных для данного столбца допустим для полнотекстового поиска и если таблица, к которой относится этот столбец, имеет соответствующий полнотекстовый индекс. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Столбец полнотекстового типа.** Выводит имя столбца, по которому построен полнотекстовый индекс для данного столбца. Это свойство можно задать, если значение свойства Тип данных для данного столбца равно либо image , либо varbinary. Указанный в этом свойстве столбец должен быть типа [n]char, [n]varchar или xml, и раскрывающийся список для этого свойства содержит только столбцы этих трех типов данных. Строки в столбце, указанном в этом свойстве, указывают тип документа соответствующих строк в столбце с возможностью полнотекстового поиска. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Язык.** Указывает язык средства разбиения по словам, использованного при индексировании этого столбца. Значение в этом свойстве — это на самом деле код локали для средства разбиения по словам. Дополнительные сведения о средствах разбиения по словам и кодах языка см. в разделе «Средства разбиения по словам и парадигматические модули». Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Статистическая семантика.** Укажите, следует ли включить статистическое семантическое индексирование для выбранного столбца. Дополнительные сведения см. в разделе Семантический поиск (SQL Server).

**Спецификация удостоверения**

Выводит данные о том, будет ли данный столбец форсировать уникальность своих значений и, если будет, то как. Значение этого свойства показывает, является ли этот столбец столбцом идентификаторов и равным по значению свойству потомка Is Identity.

**Спецификация идентификации** необходимо развернуть его и изменить свойство потомка **>Is Identity**. Свойство Is Identity показывает, является этот столбец столбцом раскрывающийся список и выберите другое значение.

**Свойство Начальное значение идентификатора** выводит начальное значение, указанное при создании данного столбца идентификаторов. Это значение присваивается первой строке таблицы. Если оставить эту ячейку пустой, по умолчанию будет присвоено значение 1. Для изменения этого свойства введите новое значение непосредственно в поле.

**Свойство Шаг приращения идентификатора** выводит значение шага идентификатора, указанное при создании данного столбца идентификаторов. Это значение — шаг приращения, добавляемый к значению Начальное значение идентификатора для каждой следующей строки. Если оставить эту ячейку пустой, по умолчанию будет присвоено значение 1. Для изменения этого свойства введите новое значение непосредственно в поле.

**Индексируемый.** Указывает, может ли столбец быть проиндексирован. Например, недетерминированные вычисляемые столбцы не могут быть проиндексированы.

**Опубликован слиянием.** Указывает, опубликован ли столбец слиянием.

**Не для репликации.** Указывает, будут ли сохранены исходные значения идентификаторов во время репликации. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Реплицировано.** Указывает, реплицирован ли данный столбец в другое место.

**RowGuid.** Указывает, будет ли SQL Server использовать столбец в качестве глобального уникального идентификатора строк таблицы RowGuid. Задать значение Да можно только для уникального столбца идентификаторов. Для редактирования свойства щелкните его значение, откройте раскрывающийся список и выберите другое значение.

**Размер.** Указывает размер в байтах, который допускается типом данных столбца. Например, тип данных nchar может иметь длину 10 (количество символов), однако для работы с кодировкой Юникод его длина будет равняться 20.