

УРОК 3

Работа с SQL Server

На этом уроке вы ознакомитесь с тем, как подключаться к серверу базы данных SQL Server и регистрироваться на нем, как выдавать команды T-SQL, создавать и заполнять таблицы, которые будут использоваться в примерах, представленных далее в этой книге.

Подключение к базе данных

Теперь, когда у вас имеется РСУБД SQL Server и клиентское программное обеспечение для работы с ней, самое время обсудить вкратце подключение к базе данных.

Аналогично всем остальным СУБД типа “клиент–сервер”, РСУБД SQL Server требует зарегистрироваться в ней перед выдачей команд. Для опознавания пользователей и их учетных данных в SQL Server поддерживается список собственных пользователей или же список пользователей Windows, которым разрешается входить в Windows, чтобы приступить к работе в этой операционной системе. Поэтому вы можете автоматически войти в систему, используя свои учетные данные в ОС Windows, или же получить приглашение на ввод регистрационного имени и пароля, что зависит от конкретной конфигурации SQL Server.

При первоначальной установке SQL Server вам, вероятнее всего, было предложено ввести регистрационное имя *системного администратора*, сокращенно обозначаемое как *sa*, а также пароль. Если вы пользуетесь собственным локальным сервером базы данных ради экспериментирования с SQL Server, то данного регистрационного имени вполне достаточно для этих целей. Но на практике регистрационное имя администратора тщательно защищено, поскольку оно дает полное право создавать таблицы, удалять целые базы данных, изменять регистрационные имена

и пароли пользователей и выполнять немало других операций в базе данных.

Чтобы подключиться к SQL Server, вам потребуется следующая информация.

- ▶ Имя хоста (т.е. имя компьютера в узле сети). Это может быть имя localhost или же имя вашего компьютера, если вы подключаетесь к локальному серверу базы данных SQL Server.
- ▶ Достоверное имя пользователя, если не используется аутентификация (т.е. его опознавание) в Windows.
- ▶ Пароль пользователя, если таковой требуется.

СОВЕТ: упрощенная локальная регистрация

Если вы пользуетесь локально установленным сервером базы данных SQL Server и инструментальным средством SQL Server Management Studio, то вам достаточно щелкнуть на кнопке Connect (Соединить), когда вы получите приглашение подключиться к базе данных.

Если же вы пользуетесь одним из клиентских приложений, обсуждавшихся на предыдущем уроке, то в нем появится диалоговое окно с приглашением ввести перечисленную выше информацию.

ПРИМЕЧАНИЕ: применение других клиентов

Если вы пользуетесь другим клиентским приложением, а не SQL Server Management Studio, вам все равно придется предоставить упомянутую выше информацию, чтобы подключиться к серверу базы данных SQL Server.

Подключившись к серверу базы данных, вы получите доступ к любым базам данных и таблицам, доступным по вашему регистрационному имени. (Вопросы регистрации, управления доступом и защиты баз данных рассматриваются на уроке 29.)

Краткое введение в SQL Server Management Studio

Изучая T-SQL, вам придется интенсивно пользоваться SQL Server Management Studio, поэтому имеет смысл вкратце ознакомиться с этим инструментальным средством. Ниже дается его краткое описание.

- ▶ У верхнего края экрана находится панель инструментов с многочисленными кнопками. Одна из этих кнопок обозначена как **New Query** (Новый запрос). Если щелкнуть на ней, откроется окно, в котором можно вводить и выполнять запросы SQL. Вам придется часто пользоваться этой кнопкой и доступным с ее помощью окном.
- ▶ Слева находится столбец с обозревателем объектов (Object Explorer), предназначенным для просмотра баз данных, таблиц и прочих элементов. Для просмотра подробных сведений об объектах, их редактирования и выполнения прочих задач нужно щелкнуть правой кнопкой мыши на одном из объектов. А для просмотра уже созданных баз данных достаточно щелкнуть на кнопке **+** рядом с меткой **Databases** (Базы данных).
- ▶ На левом краю панели инструментов, прямо над обозревателем объектов, находится раскрывающийся список (он, вероятнее всего, будет содержать слово *master* (главная) при открытии Microsoft SQL Server Management Studio в первый раз). Этот раскрывающийся список служит для выбора базы данных, с которой требуется работать (как пояснялось на уроке 1, одна СУБД позволяет работать со многими базами данных). Далее поясняется, каким образом создается база данных, которая затем появится в данном списке.
- ▶ С правой стороны на экране находится панель **Properties** (Свойства). Если она не видна, то ничего страшного — она откроется при отображении свойств.
- ▶ Панель **Results** (Результаты) изначально не видна. Она появляется под окном **Query** (Запрос) и отображает результаты запроса, как только они становятся доступными.

Пользоваться Microsoft SQL Server Management Studio можно по-разному, но для ввода и выполнения операторов SQL необходимо выполнить следующие основные действия.

- ▶ Как отмечалось выше, после щелчка на кнопке New Query в левом верхнем углу экрана открывается окно, в котором вводятся операторы SQL.
- ▶ При вводе операторов T-SQL они кодируются в Microsoft SQL Server Management Studio разным цветом, что очень удобно для поиска и исправления опечаток, недостающих кавычек и прочих ошибок.
- ▶ Чтобы выполнить оператор, щелкните на кнопке Execute (Выполнить) с изображением красного знака восклицания, или нажмите функциональную клавишу <F5>, или комбинацию клавиш <Ctrl+E>.
- ▶ Чтобы проверить правильность синтаксиса введенного оператора SQL, не выполняя его, щелкните на кнопке Parse (Выполнить синтаксический анализ) с изображением синего флажка.
- ▶ Microsoft SQL Server Management Studio отображает результаты в нижней части экрана. Результаты могут быть отображены сеткой (по умолчанию), простым текстом или сохранены в файле. Для выбора одного из этих режимов отображения достаточно щелкнуть на соответствующей кнопке панели инструментов.
- ▶ Помимо результатов выполнения операторов, Microsoft SQL Server Management Studio отображает сообщения о состоянии (например, количество строк, возвращаемых из базы данных) на второй вкладке Messages (Сообщения).
- ▶ Чтобы получить справку по отдельному оператору, щелкните на нем или нажмите функциональную клавишу <F1>.

Создание образцовых таблиц

Для проработки примеров из этой книги вам потребуются образцовые таблицы базы данных. На веб-странице, посвященной этой книге и доступной по адресу <http://www.fortacom/books/0672337924/>, находится ссылка на архивный файл,

содержащий два описываемых ниже файла сценариев SQL. Это простые текстовые файлы, содержащие операторы SQL для автоматического создания и заполнения образцовых таблиц.

- ▶ Файл `create.sql` содержит операторы T-SQL для автоматического создания шести таблиц базы данных, в том числе операторы для определения всех первичных и внешних ключей с накладываемыми на них ограничениями.
- ▶ Файл `populate.sql` содержит операторы INSERT языка SQL для заполнения упомянутых выше таблиц.

ПРИМЕЧАНИЕ: только для SQL Server

Операторы SQL в обоих загружаемых файлах сценариев с расширением `.sql` предназначены для применения только вместе с сервером базы данных Microsoft SQL Server. Сценарии были тщательно проверены на этом сервере, начиная только с версии SQL Server 2012.

ПРИМЕЧАНИЕ: сначала создать, а затем заполнить таблицы

Выполните *сначала* сценарии для создания таблиц, а *затем* сценарии для их заполнения. Проверьте любые сообщения об ошибках, возвращаемые этими сценариями. Если сценарии создания таблиц завершатся неудачно, исправьте возникшие ошибки, прежде чем перейти к заполнению таблиц.

Загрузив и распаковав архив, можете воспользоваться полученными файлами сценариев для создания и заполнения образцовых таблиц, которые потребуются для проработки уроков из этой книги. С этой целью выполните следующие действия.

1. Ради большей надежности в целях изучения T-SQL будет использоваться специально выделенная база данных. Благодаря этому исключается случайный доступ и неумышленное изменение других данных. Для этого активизируйте Microsoft SQL Server Management Studio, щелкните

правой кнопкой мыши на метке Databases в обозревателе объектов и выберите команду New Database . . . из контекстного меню, чтобы открыть диалоговое окно New Database (Новая база данных). Введите в качестве имени новой базы данных **learnsql** или что-нибудь другое, но только не имя уже существующей базы данных, чтобы перестраховаться на всякий непредвиденный случай. Можете пренебречь всеми остальными вариантами выбора в данном окне, просто щелкнув на кнопке ОК, чтобы создать базу данных.

2. Имея теперь в своем распоряжении базу данных, можете создать образцовые таблицы. Итак, щелкните на кнопке New Query, чтобы открыть окно запроса (Query Window). Выберите вновь созданную вами базу данных из раскрывающегося списка баз данных, иначе таблицы будут созданы не в той базе данных, которая вам потребуется в дальнейшем.
3. Сначала вам нужно выполнить сценарий из файла `create.sql`. Вы можете просто скопировать и вставить его содержимое в окно запроса или же открыть файл `create.sql` непосредственно по соответствующей команде, выбираемой из меню File. Как только исходный код сценария появится в окне Query Window, щелкните на кнопке Execute, чтобы выполнить сценарий и создать таблицы.
4. Повторите п.3 данной процедуры, чтобы заполнить новые таблицы по сценарию из файла `populate.sql`. И в этом случае обязательно выберите нужную базу данных.

Итак, в вашем распоряжении имеются все заполненные данными таблицы, которые потребуются для дальнейшей проработки материала этой книги.

Выбор базы данных

Как упоминалось ранее, при первом подключении к SQL Server, открывается используемая по умолчанию база данных. Обычно она называется *master* (главная), и поэтому лучше с ней

не экспериментировать. Прежде чем выполнять любые операции в базе данных, вы должны выбрать подходящую базу данных. Это можно сделать из упоминавшегося выше раскрывающегося списка на панели инструментов в окне Microsoft SQL Server Management Studio или с помощью ключевого слова `USE` языка SQL.

НОВЫЙ ТЕРМИН: *ключевое слово*

Это зарезервированное слово, являющееся составной частью синтаксиса языка T-SQL. Ключевые слова *нельзя* употреблять в качестве имен баз данных, их таблиц и столбцов. Перечень всех зарезервированных ключевых слов языка T-SQL, используемых в SQL Server, приведен в приложении Д.

Например, чтобы воспользоваться базой данных `learnsql`, введите и выполните следующую команду в окне запроса:

Ввод ▼

```
USE learnsql;
```

СОВЕТ: *развитая логика Microsoft SQL Server Management Studio*

Microsoft SQL Server Management Studio оказывает помощь в написании кода SQL. Помимо цветовой кодировки операторов SQL, это инструментальное средство пытается предложить наиболее подходящие варианты выбора, чтобы упростить написание кода и тем самым свести к минимуму ошибки и опечатки. И как только вы введете оператор `USE`, в той же самой строке появится список баз данных, доступных для выбора. Такой режим автозаполнения вводимого кода еще не раз придет вам на помощь, когда вы будете пользоваться этим инструментальным средством.

Вывод ▼

Command(s) completed successfully.

(Команды выполнены успешно)

Анализ ▼

Оператор USE не возвращает никаких результатов. Но в конечном итоге появляется сообщение, подтверждающее, что команда выполнена успешно.

СОВЕТ: выбор базы данных с помощью оператора USE или в диалоговом режиме?

Чтобы воспользоваться базой данных, ее можно выбрать из раскрывающегося списка на панели инструментов в окне SQL Server Management Studio. Так зачем же требуется оператор USE? На самом деле этот оператор автоматически выполняет одноименную команду в SQL Server Management Studio, когда вы выбираете базу данных из раскрывающегося списка, как будто бы вы ввели его в окне запроса описанным выше способом. Таким образом, вы все равно пользуетесь оператором USE, даже если выбираете базу данных из раскрывающегося списка.

Не забывайте: прежде чем получить доступ к любой информации, хранящейся в базе данных, ее следует выбрать с помощью оператора USE или из раскрывающегося списка в окне SQL Server Management Studio.

Сведения о базах данных и их таблицах

Но что если вы не знаете имен доступных баз данных? И каким образом клиентские приложения получают перечень доступных баз данных, отображаемых в раскрывающемся списке?

Сведения о базах данных, их таблицах, столбцах, пользователях, их привилегиях и прочем хранятся в таблицах самих баз

данных (для хранения этой информации SQL Server пользуется собственными ресурсами). Эти внутренние таблицы находятся в *главной* базе данных и обычно недоступны непосредственно (именно поэтому экспериментировать с главной базой данных не рекомендуется). Но для получения подобных сведений (т.е. извлечения соответствующей информации из внутренних таблиц) в SQL Server имеется набор предварительно написанных хранимых процедур.

ПРИМЕЧАНИЕ: хранимые процедуры

Более подробно хранимые процедуры рассматриваются на уроке 23. А до тех пор достаточно сказать, что хранимые процедуры состоят из операторов SQL, хранятся в SQL Server и могут быть выполнены по мере надобности.

Рассмотрим следующий пример выполнения хранимой процедуры:

Ввод ▼

```
sp_databases;
```

Вывод ▼

DATABASE_NAME	DATABASE_SIZE	REMARKS
coldfusion	9096	NULL
learnsql	3072	NULL
forta	2048	NULL
master	4608	NULL
model	1728	NULL
msdb	5824	NULL
tempdb	8704	NULL

Анализ ▼

Хранимая процедура `sp_databases` возвращает список доступных баз данных, в который могут быть включены базы

данных, предназначенные для внутреннего применения в SQL Server (в рассматриваемом здесь примере это базы данных master и tempdb). Разумеется, список доступных вам баз данных может отличаться от приведенного выше.

Чтобы получить список таблиц, находящихся в базе данных, необходимо выбрать сначала подходящую базу данных, а затем вызвать хранимую процедуру `sp_tables`, как показано ниже.

Ввод ▼

```
sp_tables;
```

Анализ ▼

Хранимая процедура `sp_tables` возвращает таблицы, находящиеся в выбранной в настоящий момент базе данных, а не только ваши собственные таблицы. К их числу относятся также всевозможные системные таблицы и прочие записи, которые могут насчитываться сотнями.

Чтобы получить список таблиц (и только их, а не представлений, системных таблиц и прочего), можно воспользоваться следующим оператором:

Ввод ▼

```
sp_tables NULL, dbo, learnsql, 'TABLE';
```

Вывод ▼

TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	TABLE_TYPE	REMARKS
crashcourse	dbo	customers	TABLE	NULL
crashcourse	dbo	orderitems	TABLE	NULL
crashcourse	dbo	orders	TABLE	NULL
crashcourse	dbo	products	TABLE	NULL
crashcourse	dbo	vendors	TABLE	NULL
crashcourse	dbo	productnotes	TABLE	NULL
crashcourse	dbo	sysdiagrams	TABLE	NULL

Анализ ▼

Здесь хранящая процедура `sp_tables` принимает ряд параметров, сообщающих ей, какой базой данных следует воспользоваться и что именно перечислить в выводимом списке. В частности, перечислить следует таблицы (параметр 'TABLE'), а не представления или системные таблицы (параметр 'VIEW' или 'SYSTEM TABLE' соответственно).

Для отображения столбцов таблицы можно воспользоваться хранимой процедурой `sp_columns` следующим образом:

Ввод ▼

```
sp_columns customers;
```

ПРИМЕЧАНИЕ: усеченный вывод ради краткости

Хранящая процедура `sp_columns` возвращает немало информации. Поэтому приведенный ниже результат ее выполнения был усечен, чтобы его можно было уместить на странице книги. Ведь иначе для вывода каждой строки таблицы потребовалась бы не одна строка текста.

Вывод ▼

TABLE_QUALIFIER	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME
learnsql	dbo	customers	cust_id	4	int identity
learnsql	dbo	customers	cust_name	-8	nchar
learnsql	dbo	customers	cust_address	-8	nchar
learnsql	dbo	customers	cust_city	-8	nchar
learnsql	dbo	customers	cust_state	-8	nchar
learnsql	dbo	customers	cust_zip	-8	nchar
learnsql	dbo	customers	cust_country	-8	nchar
learnsql	dbo	customers	cust_contact	-8	nchar
learnsql	dbo	customers	cust_email	-8	nchar

Анализ ▼

Хранящая процедура `sp_columns` требует, чтобы было указано имя таблицы (в данном примере — `customers`). Она возвращает строку таблицы, содержащую имя каждого поля (или

столбца), тип его данных, наличие пустого значения NULL, если такое допускается, сведения о ключах, значение по умолчанию и многое другое.

ПРИМЕЧАНИЕ: что такое идентичность?

Столбец `cust_id` относится к типу `identity`. В некоторых столбцах таблицы должны быть указаны однозначные значения (например, порядковые номера, идентификационные номера служащих или заказчиков, как в данном примере). Вместо того чтобы присваивать однозначные значения вручную и затем отслеживать последнее введенное значение, SQL Server может автоматически присвоить следующий доступный номер всякий раз, когда в таблицу вводится строка. Подобное свойство называется *идентичностью*. Если требуется, оно может стать частью определения таблицы, применяемого при создании таблицы с помощью оператора **CREATE**, который более подробно рассматривается на уроке 20.

В сервере базы данных SQL Server поддерживаются и другие хранимые процедуры, включая следующие:

- ▶ **sp_server_info** — служит для отображения обширной информации о состоянии сервера базы данных.
- ▶ **sp_spaceused** — предназначена для отображения объема свободного пространства, используемого (и неиспользуемого) базой данных.
- ▶ **sp_statistics** — служит для отображения статистики использования таблиц базы данных.
- ▶ **sp_helpuser** — предназначена для отображения имеющихся учетных записей пользователей.
- ▶ **sp_helplogins** — служит для отображения регистрационных данных пользователей и их прав доступа.

Следует заметить, что перечисленные выше хранимые процедуры часто применяются в клиентских приложениях. Все приложения, отображающие интерактивные списки баз данных и их таблиц, допускающие создание и редактирование таблиц в диалоговом режиме, упрощающие ввод и редактирование данных,

позволяющие управлять учетными записями и правами доступа пользователей и выполняющие другие операции, пользуются для выполнения своих функций теми же хранимыми процедурами, которые доступны и вам для непосредственного выполнения.

Резюме

На этом уроке вы научились подключаться к серверу SQL Server, выбирать базы данных с помощью оператора `USE`, а также получать сведения о внутреннем устройстве баз данных и их таблиц, используя хранимые процедуры. Вооружившись этими знаниями, смело переходите к изучению очень важного оператора `SELECT`.