

## Глава 3



# Среда управления SQL Server Management Studio

---

- ◆ Введение в среду управления SQL Server Management Studio
- ◆ Использование среды SQL Server Management Studio с компонентом Database Engine
- ◆ Разработка запросов, используя среду SQL Server Management Studio

В начале этой главе рассматривается среда управления SQL Server Management Studio, включая информацию о ее подключении к серверу, ее компонентах Registered Servers (Зарегистрированные серверы) и Object Explorer (Обозреватель объектов) и ее различных панелях пользовательского интерфейса. Далее подробно рассматриваются функциональности среды SQL Server Management Studio, связанные с компонентом Database Engine, включая возможности администрирования и управления базами данных, которые необходимо понимать, чтобы быть в состоянии создавать и выполнять инструкции языка Transact-SQL. В заключительном материале главы обсуждается использование компонентов Query Editor (Редактор запросов) и Solution Explorer (Обозреватель решений) и средства отладки для разработки запросов в среде SQL Server Management Studio.

## Введение в среду управления SQL Server Management Studio

Система SQL Server 2012 предоставляет различные инструменты для выполнения всевозможных задач, таких как установка, конфигурирование, контрольная проверка системы (аудит) и настройка ее производительности. (Инструменты для выполнения всех этих задач рассматриваются в разных главах книги.) Основным инструментом администратора для взаимодействия с системой является среда управления SQL Server Management Studio. Как администраторы, так и конечные пользователи

могут использовать этот инструмент для администрирования множественных серверов, разработки баз данных и репликации данных.



#### ПРИМЕЧАНИЕ

Эта глава полностью посвящена рассмотрению действий конечного пользователя. Поэтому в ней подробно рассматривается только функциональность среды SQL Server Management Studio применительно создания объектов баз данных посредством компонента Database Engine. Все задачи администрирования и задачи, связанные со службами анализа Analysis Services, а также другие, поддерживаемые этим инструментом компоненты рассматриваются в *части III* этой книги.

Для запуска среды SQL Server Management Studio выполните последовательность команд **Пуск | Все программы | Microsoft SQL Server 2012 | SQL Server Management Studio**.

Среда управления SQL Server Management Studio состоит из нескольких разных компонентов, которые используются для администрирования и управления всей системой. Основные из этих компонентов перечислены в следующем списке:

- ◆ Registered Servers (Зарегистрированные серверы);
- ◆ Object Explorer (Обозреватель объектов);
- ◆ Query Editor (Редактор запросов);
- ◆ Solution Explorer (Обозреватель решений).

В этом разделе рассматриваются первые два компонента списка. Редактор запросов и обозреватель решений рассматриваются далее в *разд. "Разработка запросов, используя среду SQL Server Management Studio"* этой главы.

Чтобы открыть главный интерфейс среды SQL Server Management Studio, нужно сначала подключиться к серверу, как это описывается в следующем разделе.

## Подключение к серверу

При запуске среды SQL Server Management Studio открывается диалоговое окно **Connect to Server** (Соединение с сервером) (рис. 3.1), в котором нужно задать необходимые параметры для подключения к серверу.

- ◆ **Server type** (Тип сервера). Для целей этой главы из раскрывающегося списка выберите опцию **Database Engine** (Компонент Database Engine).



#### ПРИМЕЧАНИЕ

С помощью среды SQL Server Management Studio, среди прочего, можно управлять объектами компонента Database Engine и служб Analysis Services. В этой главе рассматривается использование среды SQL Server Management Studio только для управления объектами компонента Database Engine.

- ◆ **Server name** (Имя сервера). Выберите из раскрывающегося списка или введите с клавиатуры имя сервера, к которому нужно подключиться. (Обычно, среду SQL

Server Management Studio можно подключить к любому установленному продукту на конкретном сервере.)

- ◆ **Authentication** (Проверка подлинности). Выберите один из следующих двух типов проверки подлинности:
  - **Windows Authentication** (Проверка подлинности Windows). Подключиться к SQL Server по своей учетной записи Windows. Это наиболее легкий вариант подключения и рекомендуется компанией Microsoft;
  - **SQL Server Authentication** (Проверка подлинности SQL Server). Используется проверка подлинности компонента Database Engine.

#### ПРИМЕЧАНИЕ

Дополнительную информацию, связанную с проверкой подлинности SQL Server, см. в главе 12.

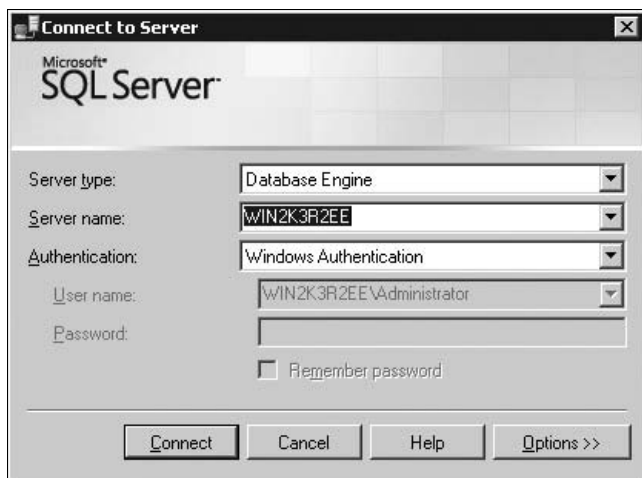


Рис. 3.1. Диалоговое окно **Connect to Server**

Указав все необходимые параметры, нажмите кнопку **Connect** (Соединить) и Database Engine подключится к указанному серверу. После подключения к серверу базы данных открывается главное окно среды SQL Server Management Studio. Своим внешним видом это окно похоже на главное окно среды разработки Visual Studio, поэтому пользователи могут применить свой опыт работы в Visual Studio в данной среде. На рис. 3.2 показано главное окно среды SQL Server Management Studio с несколькими панелями.

#### ПРИМЕЧАНИЕ

Среда SQL Server Management Studio предоставляет единый интерфейс для управления серверами и создания запросов для всех компонентов SQL Server. Иными словами, для компонентов Database Engine, служб Analysis Services, служб Integration Services и служб Reporting Services применяется один и тот же интерфейс.

## Компонент Registered Servers

Компонент Registered Servers (Зарегистрированные серверы) представлен в виде панели, позволяющей работать с уже использованными серверами (см. рис. 3.2). Если панель **Registered Servers** (Зарегистрированные серверы) отсутствует, то ее можно открыть, выбрав ее имя в меню **View** (Вид). С помощью этих подключений можно проверять состояние сервера или управлять его объектами. Для каждого пользователя применяется отдельный список зарегистрированных серверов, который хранится локально.

В список можно добавлять новые серверы или же удалять из него находящиеся в нем. Серверы можно упорядочивать по группам. В каждую такую группу следует помещать серверы, между которыми существует логическая связь. Серверы также можно группировать по типу, например серверы для компонента Database Engine, служб Analysis Services, Reporting Services и Integration Services.

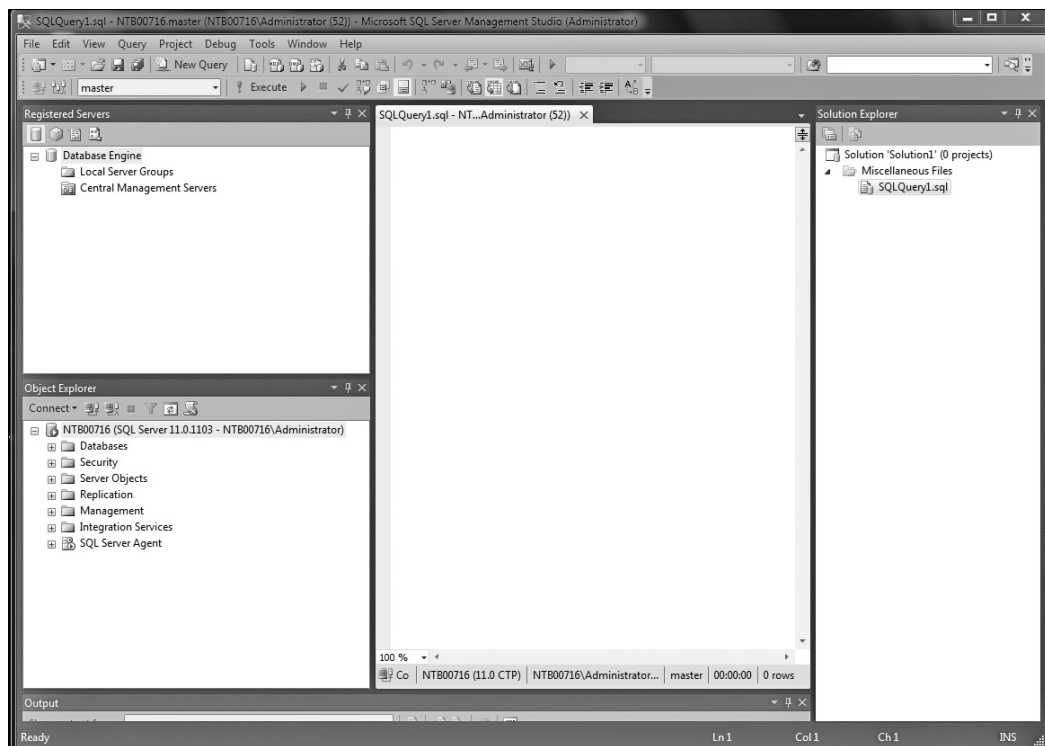


Рис. 3.2. Среда управления SQL Server Management Studio

## Компонент Object Explorer

Панель **Object Explorer** (Обозреватель объектов) содержит в виде дерева представление всех объектов баз данных сервера. Если панель **Object Explorer** (Обозреватель объектов) не отображена, то ее можно открыть, выбрав последовательность

команд из меню **View | Object Explorer** (Вид | Инспектор объектов). Данное древовидное представление отображает иерархию объектов на сервере. Таким образом, если ее развернуть, будет показана логическая структура соответствующего сервера.

Обозреватель объектов позволяет подключаться в одной панели к нескольким серверам. Это могут быть любые из имеющихся серверов для компонента Database Engine, служб Analysis Services, Reporting Services или Integration Services. Данная возможность облегчает работу пользователя, поскольку она позволяет управлять всеми серверами одного или разных типов с одного места.



#### ПРИМЕЧАНИЕ

Обозреватель объектов также обладает другими функциональностями, которые рассматриваются далее в этой главе.

## Организация панелей среды SQL Server Management Studio и перемещение по ним

Все панели среды SQL Server Management Studio можно закреплять в главном окне или скрывать из вида. Щелчок правой кнопкой мыши по строке заголовка панели предоставляет выбор из следующих вариантов отображения данной панели:

- ♦ **Floating** (Плавающая область) — панель становится свободно перемещающейся поверх остальных панелей среды SQL Server Management Studio, и ее можно поместить в любом месте на экране;
- ♦ **Dockable** (Закрепить) — панель можно переместить и закрепить в требуемом месте. Чтобы переместить панель в нужное место, щелкните ее строку заголовка и, не отпуская кнопки мыши, перетащите панель, куда следует;
- ♦ **Tabbed Document** (Закрепить как вкладки) — панели можно организовать в виде вкладок документа, когда состояние панели изменяется из закрепляемой на документ с вкладкой;
- ♦ **Hide** (Скрыть) — панель можно скрыть. Альтернативно панель можно скрыть, щелкнув крестик в ее правом верхнем углу. Чтобы снова отобразить закрытую панель, выберите ее имя в меню **View** (Вид);
- ♦ **Auto Hide** (Автоматически скрывать) — панель сворачивается и прикрепляется в виде вкладки на левой стороне экрана. Чтобы открыть (развернуть) такую панель, наведите указатель мыши на вкладки на левой стороне экрана, а чтобы удерживать панель открытой, нажмите значок канцелярской кнопки в правом верхнем углу панели.



#### ПРИМЕЧАНИЕ

Различие между режимами **Hide** (Скрыть) и **Auto Hide** (Автоматически скрывать) состоит в том, что в первом случае панель полностью убирается из представления в среде SQL Server Management Studio, а во втором она сворачивается во вкладку.

Для того чтобы восстановить конфигурацию по умолчанию, выберите последовательность команд из меню **Window | Reset Window Layout** (Окно | Сброс макета окон). После сброса настроек с левой стороны среды SQL Server Management Studio располагается панель обозревателя объектов, а с правой — вкладка **Object Explorer Details** (Подробности обозревателя объектов). На вкладке **Object Explorer Details** (Подробности обозревателя объектов) отображается информация о текущем узле, выбранном в обозревателе объектов.



#### ПРИМЕЧАНИЕ

Среда SQL Server Management Studio позволяет выполнять одну и ту же задачу несколькими способами. В этой главе рассматривается несколько способов выполнения одной задачи, но в последующих главах будет разбираться только один способ. Разные люди отдают предпочтение различным методам — некоторым более по душе двойной щелчок, другие щелкают значки "+" / "-", третьи пользуются правой кнопкой, четвертые обращаются к раскрывающимся меню, пятым нравятся ярлыки и т. п. Чтобы определить наиболее удобный для вас способ перемещения по среде, экспериментируйте с разными способами, пока не выберите самый подходящий.

Подобъект отображается в панелях **Object Explorer** (Обозреватель объектов) и **Registered Servers** (Зарегистрированные серверы) только в том случае, если щелкнуть значок плюс "+" его немедленного корневого узла в дереве иерархии. Чтобы просмотреть свойства объекта, щелкните по нему правой кнопкой мыши и в появившемся контекстном меню выберите пункт **Properties** (Свойства). Знак минус (−) слева от имени объекта означает, что иерархия данного объекта развернута. Чтобы свернуть иерархию подобъектов объекта, нужно опять щелкнуть этот значок. (Другим подходом к сворачиванию иерархии объекта будет выполнение двойного щелчка по его папке или выбор его папки и нажатие клавиши <←> "стрелка влево".)

## Использование среды SQL Server Management Studio с компонентом Database Engine

Среда SQL Server Management Studio имеет два основных назначения:

- ♦ администрирование серверов баз данных;
- ♦ управление объектами баз данных.

Эти функции рассматриваются в следующих разделах.

### Администрирование серверов баз данных

Задачи администрирования, которые можно выполнять с помощью среды SQL Server Management Studio, включают, среди прочих, следующие:

- ♦ регистрация серверов;
- ♦ подключение к серверу;

- ♦ создание новых групп серверов;
- ♦ управление множественными серверами;
- ♦ пуск и остановка серверов.

Эти задачи администрирования описываются в следующих подразделах.

## Регистрация серверов

Среда SQL Server Management Studio отделяет деятельность по регистрации серверов от деятельности по исследованию баз данных и их объектов. (Действия этих обоих типов можно выполнять посредством обозревателя объектов.) Прежде чем можно использовать базы данных и объекты любого сервера, будь то локального или удаленного, его нужно зарегистрировать. Сервер можно зарегистрировать при первом запуске среды SQL Server Management Studio или позже. Чтобы зарегистрировать сервер базы данных, щелкните правой кнопкой требуемый сервер в обозревателе объектов и в контекстном меню выберите пункт **Register** (Зарегистрировать). Если панель обозревателя объектов скрыта, то откройте ее, выбрав последовательность команд из меню **View | Object Explorer** (Вид | Обозреватель объектов). Откроется диалоговое окно **New Server Registration** (Регистрация нового сервера), как это показано на рис. 3.3.

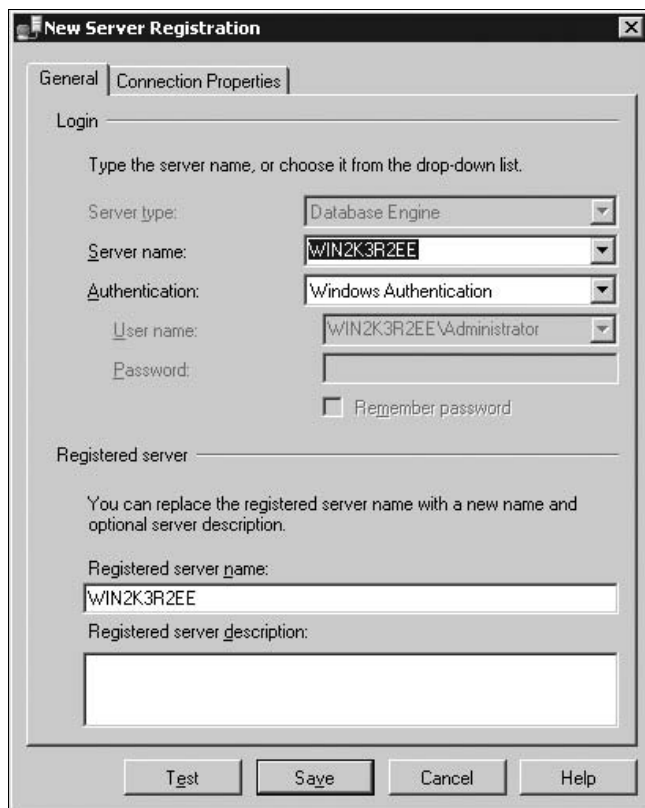


Рис. 3.3. Диалоговое окно **New Server Registration**

Выберите имя сервера, который нужно зарегистрировать, и тип проверки подлинности для этого сервера (т. е. проверка подлинности Windows или проверка подлинности SQL Server), после чего нажмите кнопку **Save** (Сохранить).

## Подключение к серверу

Среда SQL Server Management Studio также разделяет задачи регистрации сервера и подключения к серверу. Это означает, что при регистрации сервера автоматического подключения этого сервера не происходит. Чтобы подключиться к зарегистрированному серверу, нужно щелкнуть правой кнопкой требуемый сервер в окне инспектора объектов и в появившемся контекстном меню выбрать пункт **Connect** (Подключиться).

## Создание новой группы серверов

Чтобы создать новую группу серверов в панели зарегистрированных серверов, щелкните правой кнопкой узел **Local Server Groups** (Группы локальных серверов) и в контекстном меню выберите пункт **New Server Group** (Создание группы серверов). В открывшемся диалоговом окне **New Server Group Properties** (Свойства новой группа серверов) введите однозначное имя группы и, по выбору, ее описание.

## Управление множественными серверами

Посредством обозревателя объектов среда SQL Server Management Studio позволяет администрировать множественные серверы баз данных (называемые *экземплярами*) на одном компьютере. Каждый экземпляр компонента Database Server имеет свой собственный набор объектов баз данных (системные и пользовательские базы данных), который не разделяется между экземплярами.

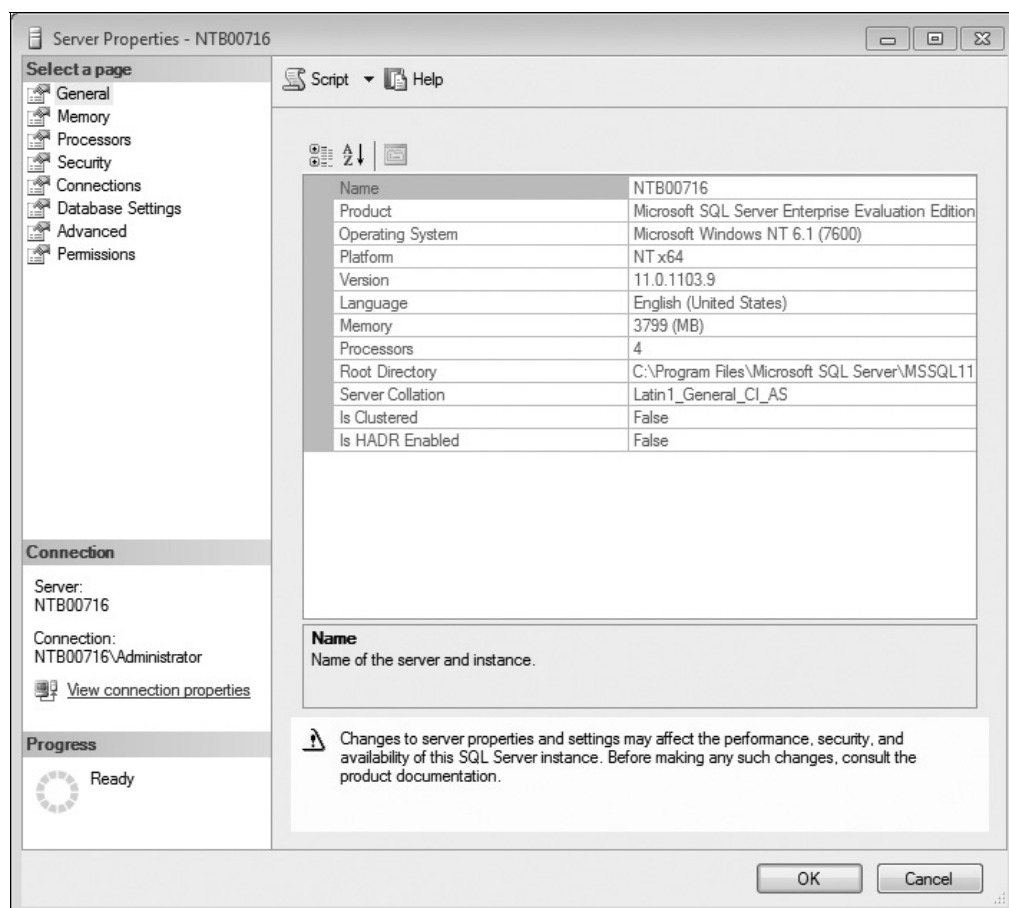
Для управления сервером и его конфигурацией щелкните правой кнопкой имя сервера в обозревателе объектов и в появившемся контекстном меню выберите пункт **Properties** (Свойства). Откроется диалоговое окно **Server Properties** (Свойства сервера), содержащее несколько страниц, таких как **General** (Общие), **Security** (Безопасность), **Permissions** (Разрешения) и т. п.

На странице **General** (Общие) (рис. 3.4) отображаются общие свойства сервера.

Страница **Security** (Безопасность) содержит информацию о режиме аутентификации сервера и методе аудита входа. На странице **Permissions** (Разрешения) воспроизводятся все учетные записи и роли, которые имеют доступ к серверу. В нижней части страницы отображаются все разрешения, которые можно предоставлять этим учетным записям и ролям.

Можно изменить имя сервера, присвоив ему новое имя. Для этого щелкните правой кнопкой требуемый сервер в окне обозревателя объектов и в контекстном меню выберите пункт **Register** (Зарегистрировать). Теперь можно присвоить серверу новое имя и изменить его описание.



Рис. 3.4. Страница **General** диалогового окна **Server Properties****ПРИМЕЧАНИЕ**

Серверы не следует переименовывать без особой на это надобности, поскольку это может повлиять на другие серверы, которые ссылаются на них.

**Запуск и останов серверов**

Сервер Database Engine по умолчанию запускается автоматически при запуске операционной системы Windows. Чтобы запустить сервер с помощью среды SQL Server Management Studio, щелкните правой кнопкой требуемый сервер в инспекторе объектов и в контекстном меню выберите пункт **Start** (Запустить). Это меню также содержит пункты **Stop** (Остановить) и **Pause** (Приостановить) для выполнения соответствующих действий с сервером.

## Управление базами данных посредством обозревателя объектов

Задачи администрирования, которые можно выполнять с помощью среды SQL Server Management Studio, включают, среди прочих, следующие:

- ♦ создание баз данных, не прибегая к использованию языка Transact-SQL;
- ♦ модифицирование баз данных, не прибегая к использованию языка Transact-SQL;
- ♦ управление таблицами, не прибегая к использованию языка Transact-SQL;
- ♦ создание и исполнение инструкций SQL (описывается далее в этой главе в разд. "Редактор запросов".)

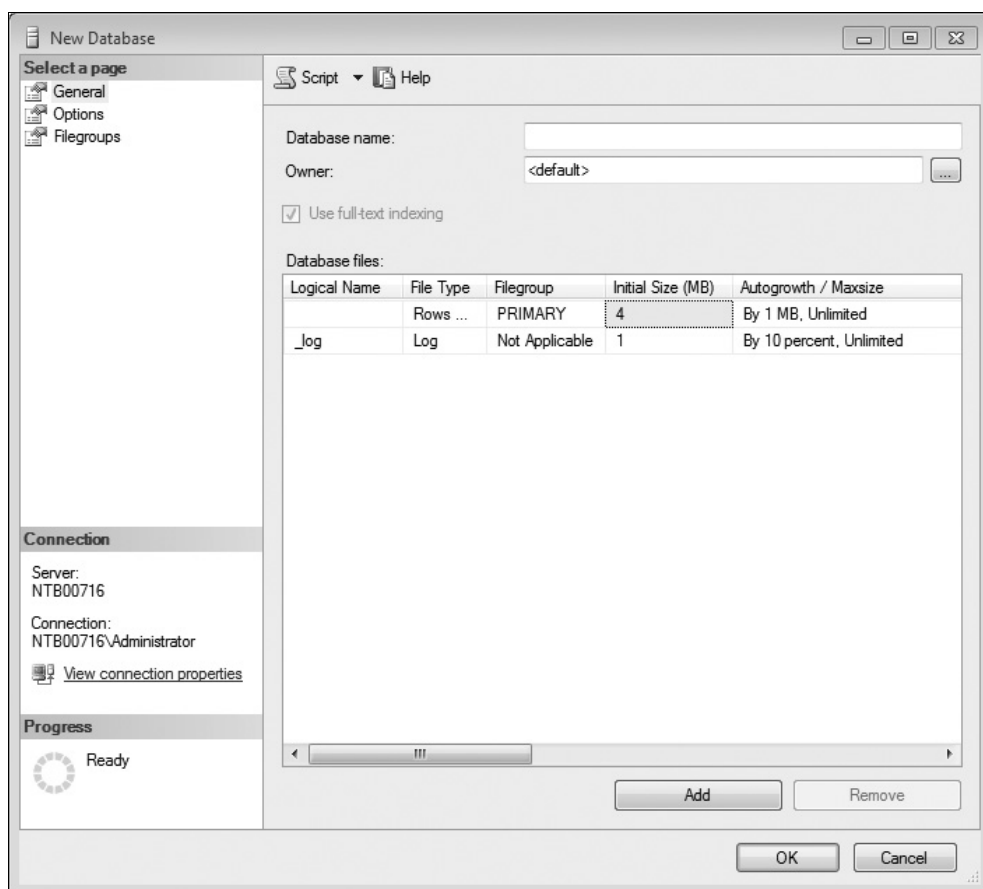
### Создание баз данных, не прибегая к использованию языка Transact-SQL

Новую базу данных можно создать посредством обозревателя объектов Object Explorer или языка Transact-SQL. (Создание баз данных с помощью языка Transact-SQL рассматривается в *главе 5*.) Как можно судить по его названию, обозреватель объектов также можно использовать для исследования объектов сервера. С панели этого инструмента можно просматривать все объекты сервера и управлять сервером и базами данных. Дерево иерархии объектов сервера содержит, среди прочих папок, папку **Databases** (Базы данных). Эта папка, в свою очередь, содержит несколько подпапок, включая папку для системных баз данных, и по папке для каждой базы данных, созданной пользователем. (Системные и пользовательские базы данных подробно рассматриваются в *главе 15*.)

Чтобы создать базу данных посредством обозревателя объектов, щелкните правой кнопкой узел **Databases** (Базы данных) и выберите пункт меню **New Database** (Создать базу данных). В открывшемся диалоговом окне **New Database** (Создание базы данных) (рис. 3.5) в поле **Database name** (Имя базы данных) введите имя новой базы данных, после чего нажмите кнопку **OK**.

Каждая база данных обладает несколькими свойствами, такими как тип файла, начальный размер и т. п. Список страниц свойств базы данных расположен в левой панели диалогового окна **New Database** (Создание базы данных). Существует несколько разных страниц (групп) свойств:

- ♦ **General** (Общие);
- ♦ **Files** (Файлы);
- ♦ **Filegroups** (Файловые группы);
- ♦ **Options** (Параметры);
- ♦ **Change Tracking** (Отслеживание изменений);
- ♦ **Permissions** (Разрешения);
- ♦ **Extended Properties** (Расширенные свойства);

Рис. 3.5. Диалоговое окно **New Database**

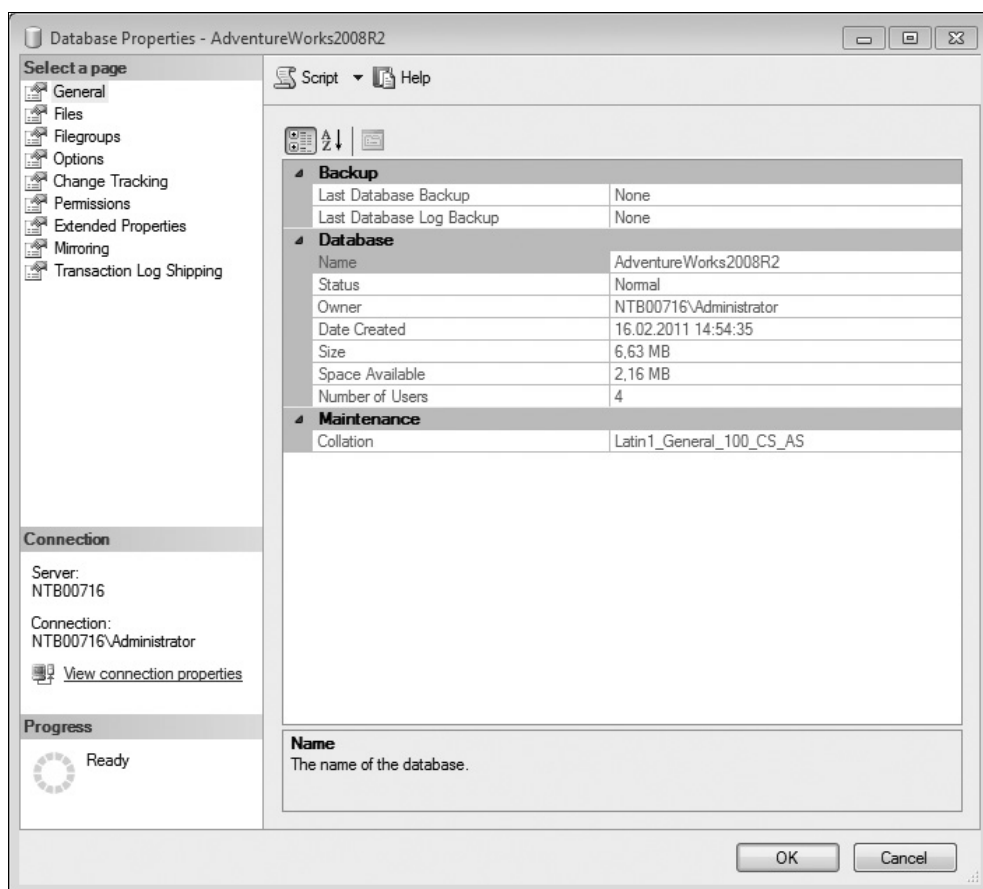
- ◆ **Mirroring** (Зеркальное отображение);
- ◆ **Transaction Log Shipping** (Доставка журналов транзакций).

**ПРИМЕЧАНИЕ**

Для уже существующей базы данных отображаются все группы свойств в ранее перечисленном списке. Для создаваемой базы данных существует только три страницы свойств: **General** (Общие), **Options** (Параметры) и **Filegroups** (Файловые группы) (см. рис. 3.5).

Страница **General** (Общие) диалогового окна **Database Properties** (Свойства базы данных) (рис. 3.6) содержит, среди прочего, такую информацию, как имя, владелец и параметры сортировки базы данных.

Свойства файлов данных определенной базы данных перечисляются на странице **Files** (Файлы) и содержат такую информацию, как имя и начальный размер файла, расположение базы данных, а также тип файла (например, **PRIMARY**). База данных может храниться в нескольких файлах.

Рис. 3.6. Страница **General** диалогового окна **Database Properties**

## ПРИМЕЧАНИЕ

В SQL Server применяется динамическое управление дисковым пространством. Это означает, что можно сконфигурировать размер базы данных для автоматического увеличения и уменьшения по мере надобности. Чтобы изменить свойство **Autogrowth** (Автоувеличение) на странице **Files** (Файлы), в столбце **Autogrowth** (Автоувеличение) нажмите значок троеточия (...) и внесите соответствующие изменения в диалоговом окне **Change Autogrowth** (Изменить авторасширение). Чтобы позволить автоматическое увеличение размера базы данных, нужно установить флажок **Enable Autogrowth** (Разрешить авторасширение). Каждый раз, когда существующий размер файла недостаточен для хранения добавляемых данных, сервер автоматически запрашивает систему выделить файлу дополнительное дисковое пространство. Объем дополнительного дискового пространства (в процентах или мегабайтах) указывается в поле **File Growth** (Увеличение размера файла) в том же диалоговом окне. А в разделе **Maximum File Size** (Максимальный размер файла) можно или ограничить максимальный размер файла, установив переключатель **Limited to (MB)** (Ограниченное (Мбайт)), или снять ограничения на размер, установив переключатель **Unlimited** (Без ограничений) (это настройка по умолчанию). При ограниченном размере файла нужно указать его допустимый максимальный размер.

На странице **Filegroups** (Файловые группы) диалогового окна **Database Properties** (Свойства базы данных) отображаются имена файловых групп, к которым принадлежит файл базы данных, раздел файловой группы (по умолчанию или заданный явно), а также операции, разрешенные для выполнения с файловой группой (чтение и запись или только чтение).

На странице **Options** (Параметры) диалогового окна **Database Properties** (Свойства базы данных) можно просмотреть и модифицировать все параметры уровня базы данных. Существуют следующие группы параметров: **Automatic** (Автоматически), **Containment** (Включение), **Cursor** (Курсор), **Miscellaneous** (Вспомогательные), **Recovery** (Восстановление), **Service Broker** (Компонент Service Broker) и **State** (Состояние).

Группа **State** (Состояние) содержит, например, следующие четыре параметра.

- ◆ **Database Read-Only** (База данных доступна только для чтения). Позволяет установить доступ к базе данных полный доступ или доступ только для чтения. В последнем случае пользователи не могут модифицировать данные. Значение по умолчанию этого параметра — `False`.
- ◆ **Restrict Access** (Ограничение доступа). Устанавливает количество пользователей, которые могут одновременно использовать базу данных. Значение по умолчанию — `MULTI_USER`.
- ◆ **Database State** (Состояние базы данных). Описывает состояние базы данных. Значение по умолчанию этого параметра — `Normal`.
- ◆ **Encryption Enabled** (Шифрование включено). Определяет режим шифрования базы данных. Значение по умолчанию этого параметра — `False`.

На странице **Extended Properties** (Расширенные свойства) отображаются дополнительные свойства текущей базы данных. На этой странице можно удалять существующие свойства и добавлять новые.

На странице **Permissions** (Разрешения) отображаются все пользователи, роли и соответствующие разрешения. (Тема разрешений подробно рассматривается в главе 12.)

Остальные страницы **Change Tracking** (Отслеживание изменений), **Mirroring** (Зеркальное отображение) и **Transaction Log Shipping** (Доставка журналов транзакций) описывают возможности, связанные с доступностью данных, и поэтому рассматриваются в главе 16.

## Модифицирование баз данных, не прибегая к использованию языка Transact-SQL

С помощью обозревателя объектов можно модифицировать существующие базы данных, изменяя файлы и файловые группы базы данных. Чтобы добавить новые файлы в базу данных, щелкните правой кнопкой требуемую базу данных и в контекстном меню выберите пункт **Properties** (Свойства). В открывшемся диалоговом окне **Database Properties** (Свойства базы данных) выберите страницу **Files** (Файлы) и нажмите кнопку **Add** (Добавить), расположенную внизу раздела **Database files**

(Файлы базы данных). В раздел будет добавлена новая строка, в поле **Logical Name** (Логическое имя) которой следует ввести имя добавляемого файла базы данных, а в других полях задать необходимые свойства этого файла. Также можно добавить и вторичную файловую группу для базы данных, выбрав страницу **Filegroups** (Файловые группы) и нажав кнопку **Add** (Добавить).



#### ПРИМЕЧАНИЕ

Упомянутые ранее свойства базы данных может модифицировать только системный администратор или владелец базы данных.

Чтобы удалить базы данных с помощью обозревателя объектов, щелкните правой кнопкой имя требуемой базы данных и в открывшемся контекстном меню выберите пункт **Delete** (Удалить).

## Управление таблицами, не прибегая к использованию языка Transact-SQL

Следующей задачей после создания базы данных является создание всех необходимых таблиц. Подобно созданию базы данных, таблицы в ней также можно создать либо с помощью языка Transact-SQL, либо посредством обозревателя объектов. Как и в случае с созданием базы данных, здесь мы рассмотрим создание таблиц только с помощью обозревателя объектов. (Создание таблиц и всех других объектов баз данных посредством языка Transact-SQL подробно рассматривается в главе 5.)

Для практики создания таблиц, в базе данных `sample` создадим таблицу `department`. Чтобы создать таблицу базы данных с помощью обозревателя объектов, разверните в нем узел **Databases** (Базы данных), а потом узел требуемой базы данных, щелкните правой кнопкой папку **Tables** (Таблицы) и в открывшемся контекстном меню выберите пункт **New Table** (Создать таблицу). В верхней части с правой стороны окна средства Management Studio откроется окно для создания столбцов новой таблицы. Введите имена столбцов таблицы, их типы данных и разрешение значений `NULL` для каждого столбца, как это показано в правой верхней панели на рис. 3.7.

Чтобы выбрать для столбца один из поддерживаемых системой типов данных, в столбце **Data Type** (Тип данных) выберите, а затем нажмите направленный вниз треугольник у правого края поля (этот треугольник появляется после того, как будет выбрана ячейка). В результате в открывшемся раскрывающемся списке выберите требуемый тип данных для столбца.

Тип данных существующего столбца можно изменить на вкладке **Column Properties** (Свойства столбца) (нижняя панель справа на рис. 3.7). Для одних типов данных, таких как `char`, требуется указать длину в строке **Length** (Длина), а для других, таких как `decimal`, на вкладке **Column Properties** (Свойства столбца) требуется указать масштаб и точность в соответствующих строках **Scale** (Масштаб) и **Precision** (Точность). Для некоторых других, таких как `int`, не требуется указывать

ни одно из этих свойств. (Недействительные значения для конкретного типа данных выделены затененным шрифтом в списке всех возможных свойств столбца.)

Чтобы разрешить значения `NULL` для данного столбца, следует установить для него соответствующий флажок поля. Также, если для столбца требуется значение по умолчанию, его следует ввести в строку **Default Value or Binding** (Значение по умолчанию или привязка) панели **Column Properties** (Свойства столбца). Значение по умолчанию присваивается ячейке столбца автоматически, если для нее явно не введено значение.

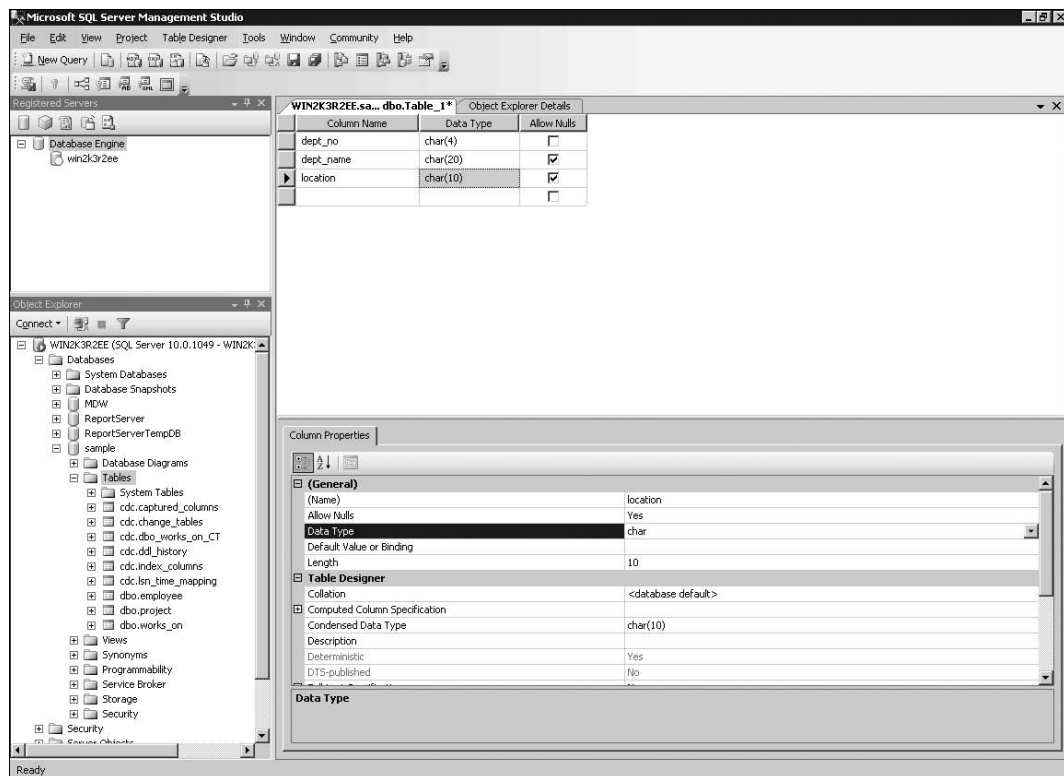


Рис. 3.7. Создание таблицы department базы данных sample посредством обозревателя объектов

Столбец `dept_no` является первичным ключом таблицы `department`. (Подробное обсуждение темы первичных ключей базы данных `sample` см. в главе 1). Чтобы сделать столбец первичным ключом таблицы, щелкните его правой кнопкой и в контекстном меню выберите пункт **Set Primary Key** (Задать первичный ключ). Завершив все работы по созданию таблицы, щелкните крестик вкладки конструктора таблиц. Откроется диалоговое окно с запросом, сохранить ли сделанные изменения. Нажмите кнопку **Yes** (Да), после чего откроется диалоговое окно **Choose Name** (Выбор имени) с запросом ввести имя таблицы. Введите требуемое имя таблицы и нажмите кнопку **OK**. Таблица будет сохранена под указанным именем. Чтобы ото-

бразить новую таблицу в иерархии базы данных, в панели инструментов обозревателя объектов щелкните значок **Renew** (Обновить).

Для просмотра и изменения свойств существующей таблицы разверните узел базы данных, содержащей требуемую таблицу, разверните узел **Tables** (Таблицы) в этой базе данных и щелкните правой кнопкой требуемую таблицу, а затем в контекстном меню выберите пункт **Properties** (Свойства). В результате для данной таблицы откроется диалоговое окно **Table Properties** (Свойства таблицы). Для примера, на рис. 3.8 показано диалоговое окно **Table Properties** (Свойства таблицы) на вкладке **General** (Общие) для таблицы `employee` базы данных `sample`.

Чтобы переименовать таблицу, в папке **Tables** (Таблицы) щелкните ее правой кнопкой в списке таблиц и в контекстном меню выберите пункт **Rename** (Переименовать). А чтобы удалить таблицу, щелкните ее правой кнопкой и выберите пункт **Delete** (Удалить).



### ПРИМЕЧАНИЕ

На данном этапе следует создать три остальные таблицы базы данных `sample`.

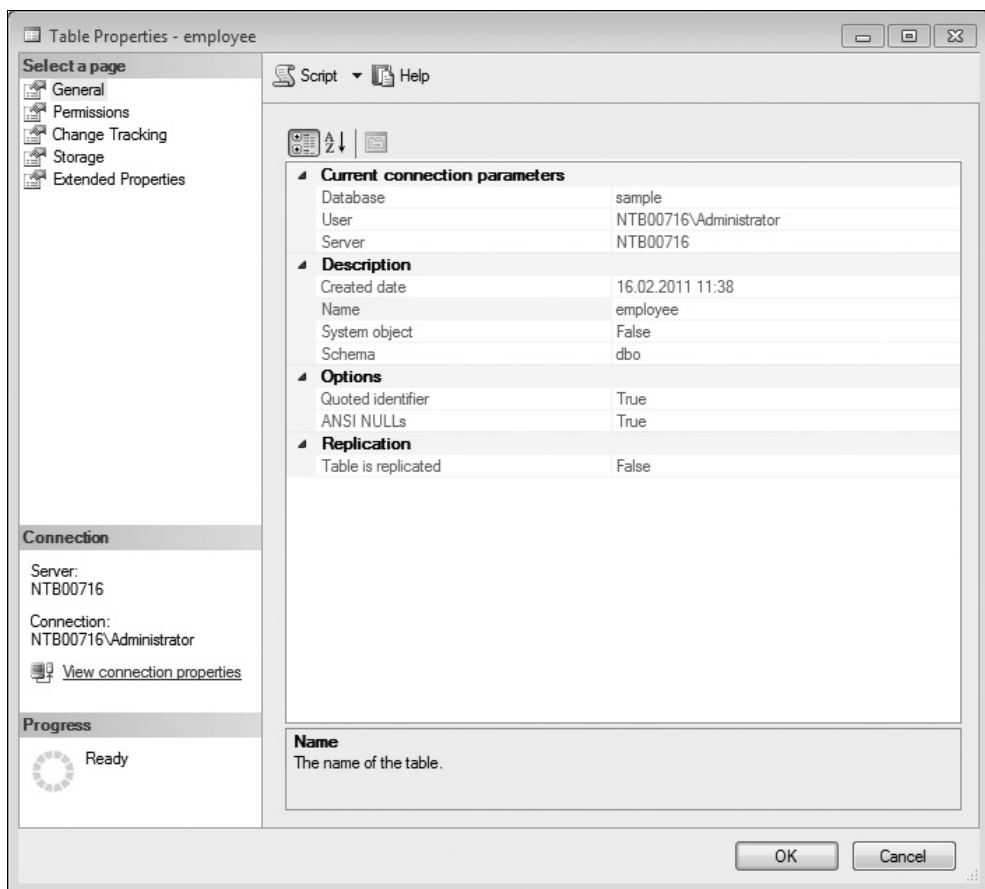


Рис. 3.8. Диалоговое окно **Table Properties**, вкладка **General** для таблицы `employee`



Создав все четыре таблицы базы данных `sample` (`employee`, `department`, `project` и `works_on`), можно использовать еще одну возможность среды SQL Server Management Studio, чтобы отобразить диаграмму типа "сущность — отношение" — диаграмму (ER) (entity-relationship) этой базы данных. (Процесс преобразования таблиц базы данных в диаграмму "сущность — отношение" (ER) называется *обратным проектированием*.)

Чтобы создать диаграмму "сущность — отношение" (ER) для базы данных `sample`, щелкните правой кнопкой ее подпапку **Database Diagrams** (Диаграммы баз данных) и в контекстном меню выберите пункт **New Database Diagram** (Создать диаграмму базы данных).



### ПРИМЕЧАНИЕ

Если откроется диалоговое окно, в котором спрашивается, создавать ли вспомогательные объекты, выберите ответ **Yes** (Да).

Откроется диалоговое окно **Add Table** (Добавление таблицы), в котором нужно выбрать таблицы для добавления в диаграмму. Добавив все необходимые таблицы (в данном случае все четыре), нажмите кнопку **Close** (Заккрыть), и мастер создаст диаграмму, подобную показанной на рис. 3.9.

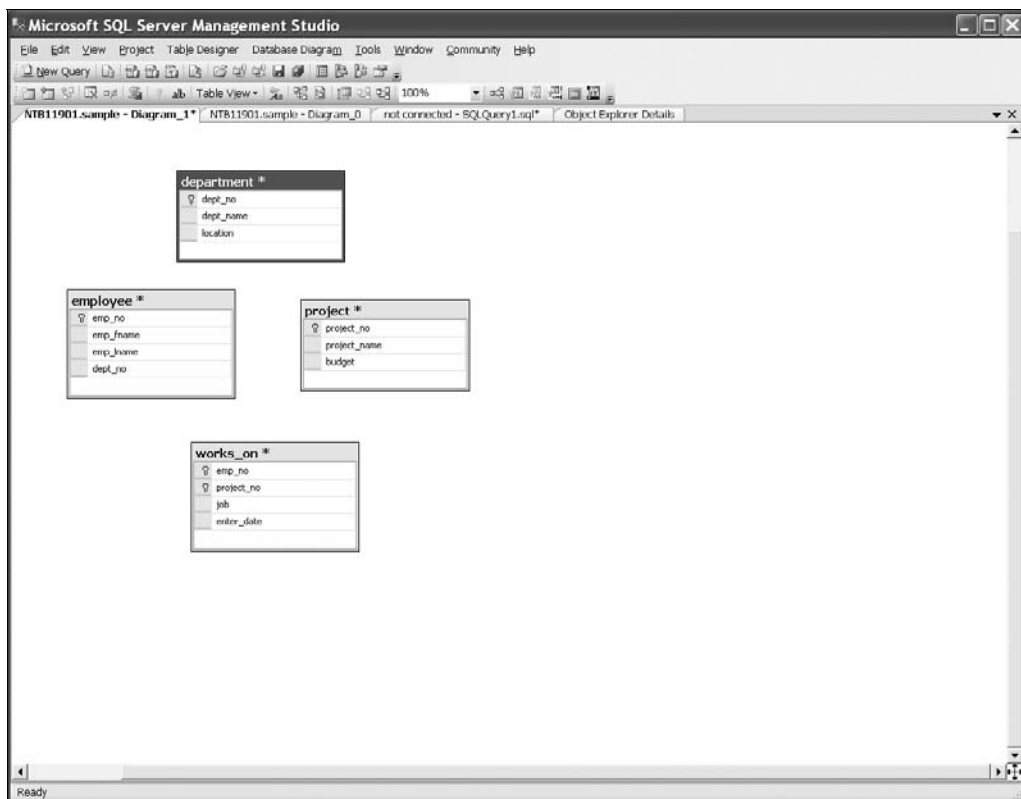


Рис. 3.9. Начальная диаграмма "сущность — отношение" (ER) базы данных `sample`

На рис. 3.9 показана только промежуточная, а не конечная, диаграмма "сущность — отношение" (ER) базы данных `sample`, поскольку, хотя на ней и показаны все четыре таблицы с их столбцами (и соответствующими первичными ключами), на ней все же отсутствуют отношения между таблицами. Отношение между двумя таблицами основывается на первичном ключе одной из таблиц и возможным соответствующим столбцом (или столбцами) другой таблицы. (Подробное обсуждение темы этих отношений и целостности ссылочных данных см. в главе 5.)

Между таблицами базы данных `sample` существует три отношения. Таблица `department` имеет отношение типа 1:N с таблицей `employee`, поскольку каждому значению первичного ключа таблицы `department` (столбец `dept_no`) соответствует одно или более значений столбца `dept_no` таблицы `employee`. Аналогично существует отношение между таблицами `employee` и `works_on`, поскольку только значения, которые присутствуют в столбце первичного ключа таблицы `employee` (`emp_no`) также имеются в столбце `emp_no` таблицы `works_on`. Третье отношение существует между таблицами `project` и `works_on`, т. к. только значения, которые присутствуют в первичном ключе таблицы `project` (`project_no`) также присутствуют в столбце `project_no` таблицы `works_on`.

Чтобы создать эти три отношения, диаграмму "сущность — отношение" (ER) нужно реконструировать, указав для каждой таблицы столбцы, которые соответствуют ключевым столбцам других таблиц. Такой столбец называется *внешним ключом* (foreign key). Чтобы увидеть, как это делается, определим столбец `dept_no` таблицы `employee`, как внешний ключ таблицы `department`. Для этого выполним следующие действия:

1. В созданной диаграмме щелкните правой кнопкой графическое представление таблицы `employee` и в контекстном меню выберите пункт **Relationships** (Отношения). В открывшемся диалоговом окне **Foreign Key Relationships** (Связи по внешнему ключу) нажмите кнопку **Add** (Добавить).
2. В правой панели диалогового окна расширьте первый столбец, выберите в нем строку **Table and Columns Specification** (Спецификация таблиц и столбцов) и нажмите кнопку с троеточием во втором столбце этой строки.
3. В открывшемся диалоговом окне **Tables and Columns** (Таблицы и столбцы) в раскрывающемся списке **Primary key table** (Таблица первичного ключа) выберите таблицу с соответствующим первичным ключом. В данном случае это будет таблица `department`.
4. Выберите для этой таблицы столбец `dept_no` в качестве первичного ключа и этот же столбец для таблицы `employee` в качестве внешнего ключа, после чего нажмите кнопку **OK**, чтобы закрыть окно **Tables and Columns** (Таблицы и столбцы). Нажмите кнопку **Close** (Заккрыть), чтобы закрыть окно **Foreign Key Relationships** (Связи по внешнему ключу).

Подобным образом создаются и другие два отношения. На рис. 3.10 показана диаграмма "сущность — отношение" (ER), отображающая все три отношения между таблицами базы данных `sample`.

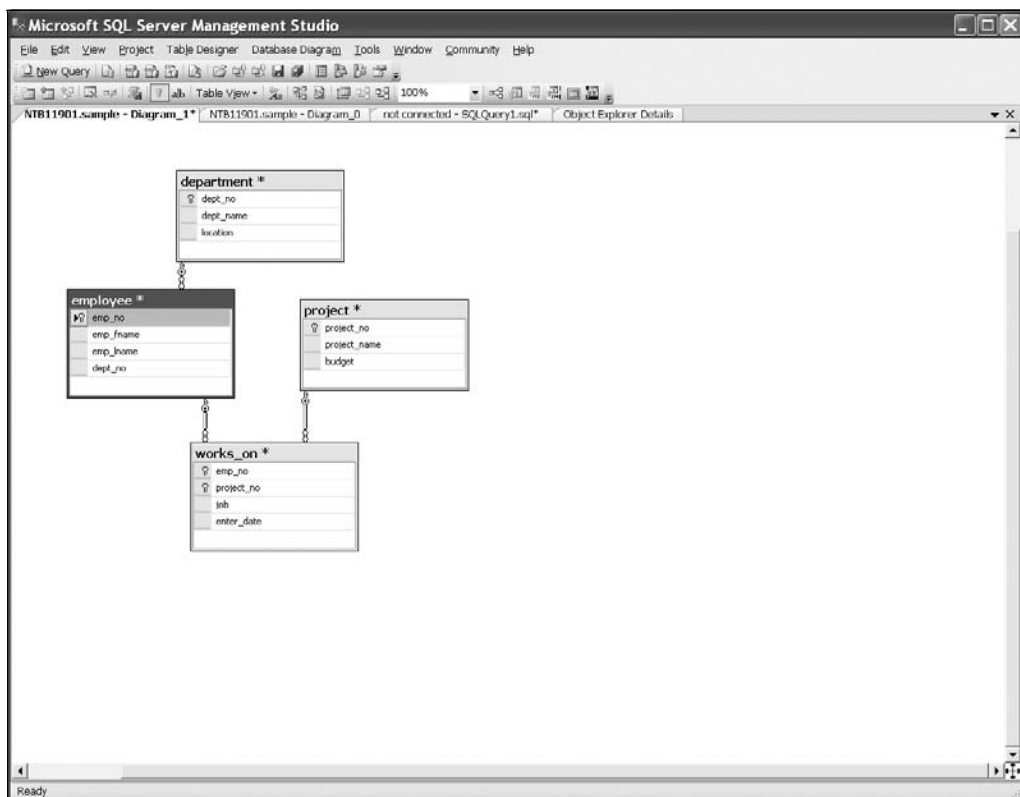


Рис. 3.10. Конечная диаграмма "сущность — отношение" (ER) базы данных sample

## Разработка запросов, используя среду SQL Server Management Studio

Среда SQL Server Management Studio предоставляет завершенное средство для создания всех типов запросов. С ее помощью можно создавать, сохранять, загружать и редактировать запросы. Кроме этого, над запросами можно работать без подключения к какому-либо серверу. Этот инструмент также предоставляет возможность разрабатывать запросы для разных проектов.

Предоставляется возможность работать с запросами как посредством редактора запросов, так и с помощью обозревателя решений. В этом разделе рассматриваются оба эти инструмента. Кроме этих двух компонентов среды SQL Server Management Studio мы рассмотрим отладку SQL-кода, используя встроенный отладчик.

### Редактор запросов

Чтобы открыть панель редактора запросов **Query Editor** (Редактор запросов), на панели инструментов среды SQL Server Management Studio нажмите кнопку **New Query** (Создать запрос). Эту панель можно расширить, чтобы отображать кнопки

создания всех возможных запросов, а не только запросов компонента Database Engine. По умолчанию создается новый запрос компонента Database Engine, но, нажав соответствующую кнопку на панели инструментов, можно также создавать запросы MDX, XMLA и др.

Строка состояния внизу панели редактора запросов указывает статус подключения редактора к серверу. Если подключение к серверу не выполнено автоматически, при запуске редактора запросов выводится диалоговое окно подключения к серверу (см. рис. 3.1), в котором можно выбрать сервер для подключения и режим проверки подлинности.



#### ПРИМЕЧАНИЕ

Редактирование запросов в автономном режиме предоставляет больше гибкости, чем при подключении к серверу. Для редактирования запросов не обязательно подключаться к серверу, и окно редактора запросов можно отключить от одного сервера (выбрав последовательность команд из меню **Query | Connection | Disconnect** (Запрос | Соединение | Отключить) и подключить к другому, не открывая другого окна редактора. Чтобы выбрать автономный режим редактирования, в диалоговом окне подключения к серверу, открывающемся при запуске редактора конкретного вида запросов, просто нажмите кнопку **Cancel** (Отмена).

Редактор запросов можно использовать для выполнения следующих задач:

- ◆ создания и выполнения инструкций языка Transact-SQL;
- ◆ сохранения созданных инструкций языка Transact-SQL в файл;
- ◆ создания и анализа планов выполнения общих запросов;
- ◆ графического иллюстрирования плана выполнения выбранного запроса.

Редактор запросов содержит встроенный текстовый редактор и панель инструментов с набором кнопок для разных действий. Главное окно редактора запросов разделено по горизонтали на панель запросов (вверху) и панель результатов (внизу). Инструкции Transact-SQL (т. е. запросы) для исполнения вводятся в верхнюю панель, а результаты обработки системой этих запросов отображаются в нижней панели. На рис. 3.11 показан пример ввода запроса в редактор запросов и результатов выполнения этого запроса.

В первой инструкции запроса **USE** указывается использовать базу данных `sample` в качестве текущей базы данных. Вторая инструкция — **SELECT** — извлекает все строки таблицы `works_on`. Чтобы выполнить этот запрос и вывести результаты, в панели инструментов редактора запросов нажмите кнопку **Execute** (Выполнить) или клавишу <F5>.



#### ПРИМЕЧАНИЕ

Можно открыть несколько окон редактора запросов, т. е. выполнить несколько подключений к одному или нескольким экземплярам компонента Database Engine. Новое подключение создается нажатием кнопки **New Query** (Создать запрос) в панели инструментов среды SQL Server Management Studio.

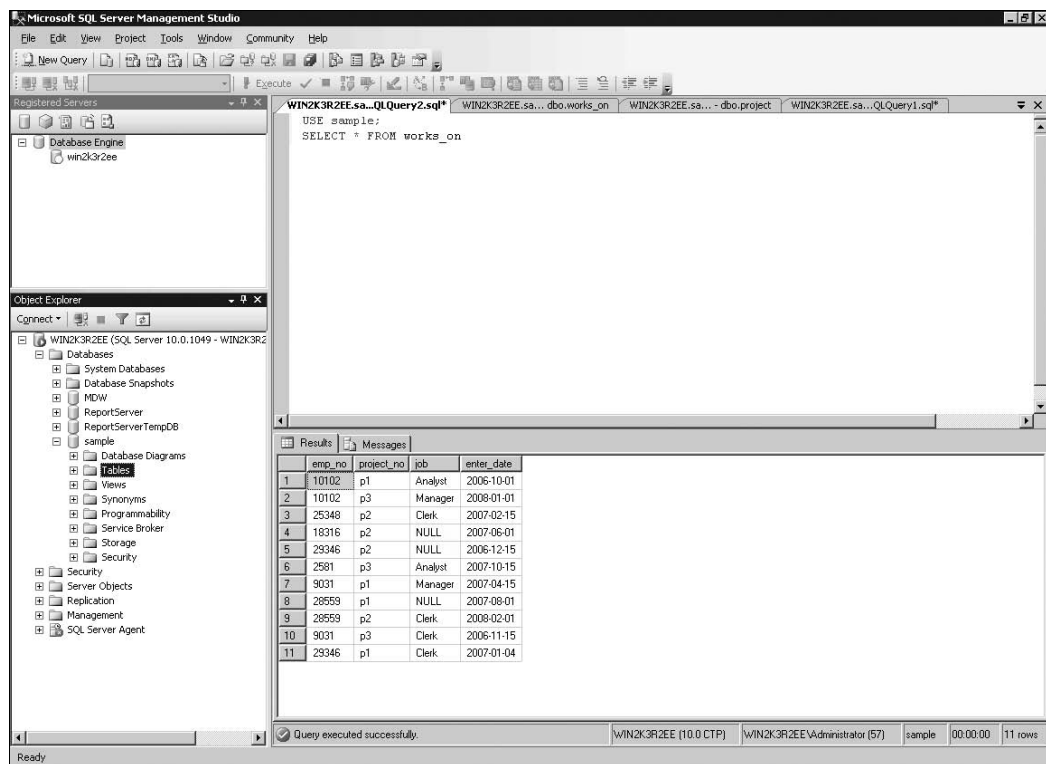


Рис. 3.11. Редактор запросов с запросом (верхняя панель) и результатами его выполнения (нижняя панель)

В строке состояния внизу окна редактора запросов отображается следующая информация, связанная с выполнением инструкций запроса:

- ◆ состояние текущей операции (например, "Запрос успешно выполнен");
- ◆ имя сервера базы данных;
- ◆ имя текущего пользователя и идентификатор серверного процесса;
- ◆ имя текущей базы данных;
- ◆ время, затраченное на выполнение последнего запроса;
- ◆ количество найденных строк.

Одним из основных достоинств среды SQL Server Management Studio является легкость ее использования, что также относится и к редактору запросов Query Editor. Редактор запросов поддерживает множество возможностей, облегчающих задачу кодирования инструкций языка Transact-SQL. В частности, в нем используется подсветка синтаксиса, чтобы улучшить читаемость инструкций языка Transact-SQL. Все зарезервированные слова отображаются синим цветом, переменные — черным, строки — красным, а комментарии — зеленым. (Зарезервированные слова рассматриваются в главе 4.)

Кроме этого, редактор запросов оснащен контекстно-зависимой справкой, называемой *Dynamic Help*, посредством которой можно получить сведения о кон-

кретной инструкции. Если вы не знаете синтаксиса инструкции, выделите ее в редакторе, а потом нажмите клавишу <F1>. Также можно выделить параметры различных инструкций Transact-SQL, чтобы получить справку по ним из электронной документации.



### ПРИМЕЧАНИЕ

В SQL Server 2012 поддерживается инструмент *SQL Intellisense*, который является видом средства автозавершения. Иными словами, этот модуль предлагает наиболее вероятное завершение частично введенных элементов инструкций Transact-SQL.

С редактированием запросов может также помочь обозреватель объектов. Например, если вы хотите узнать, как создать инструкцию `CREATE TABLE` для таблицы `employee`, щелкните правой кнопкой эту таблицу в обозревателе объектов и в появившемся контекстном меню выберите пункты **Script Table As | CREATE to | New Query Editor Window** (Создать скрипт для таблицы | Используя CREATE | Новое окно редактора запросов). Окно редактора запросов, содержащее созданную таким образом инструкцию `CREATE TABLE`, показано на рис. 3.12. Эта возможность также применима и с другими объектами, такими как хранимые процедуры и функции.

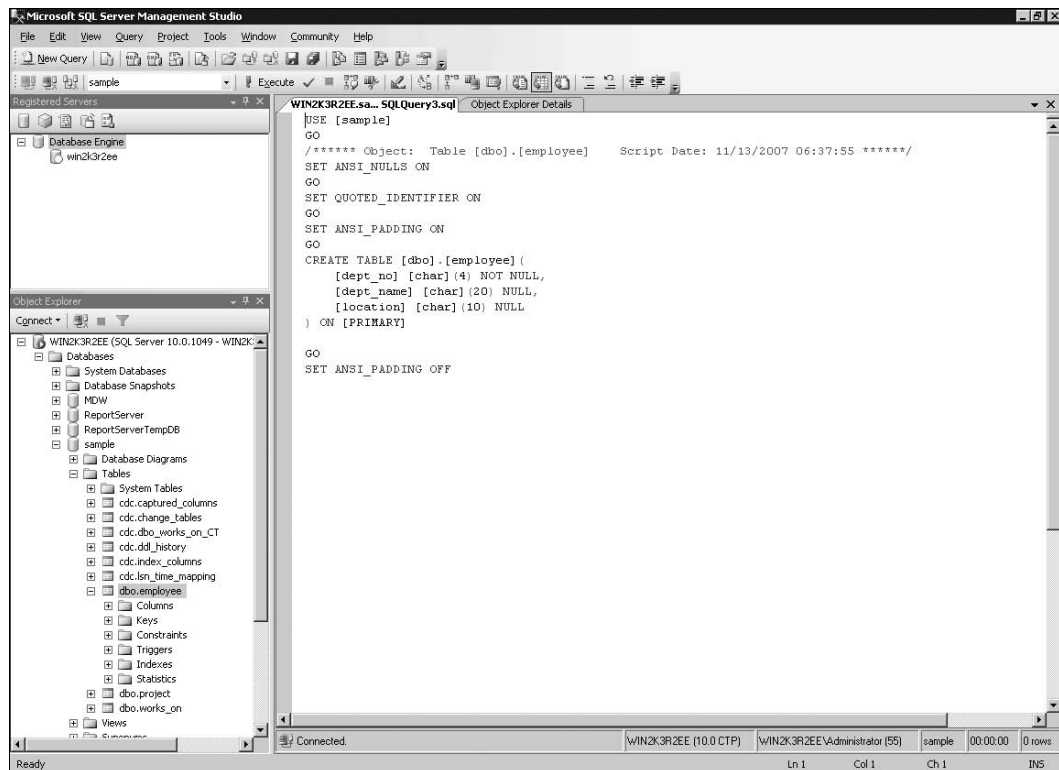


Рис. 3.12. Окно редактора запросов, содержащее созданную в нем инструкцию `CREATE TABLE`

Обозреватель объектов очень полезен для графического отображения плана исполнения конкретного запроса. *Планом выполнения запроса* называется вариант выполнения, выбранный оптимизатором запроса среди нескольких возможных вариантов выполнения конкретного запроса. Введите в верхнюю панель редактора требуемый запрос, выберите последовательность команд из меню **Query | Display Estimated Execution Plan** (Запрос | Показать предполагаемый план выполнения) и в нижней панели окна редактора будет показан план выполнения данного запроса. Эта тема подробно рассматривается в *главе 19*.

## Обозреватель решений

Редактирование запросов в среде SQL Server Management Studio основано на методе решений. Если создать пустой запрос с помощью кнопки **New Query** (Создать запрос), то он будет основан на пустом решении. Это можно увидеть, выполнив последовательность команд из меню **View | Solution Explorer** (Вид | Обозреватель решений) сразу же после открытия пустого запроса.

Решение может быть связано ни с одним, с одним или с несколькими проектами. Пустое решение, не связано ни с каким проектом. Чтобы связать проект с решением, закройте пустое решение, обозреватель решений и редактор запросов и создайте новый проект, выполнив последовательность команд из меню **File | New | Project** (Файл | Создать | Проект). В открывшемся окне **New Project** (Создать проект) выберите в средней панели опцию **SQL Server Scripts** (Скрипты SQL Server). *Проект* — это способ организации файлов в определенном месте. Проекту можно присвоить имя и выбрать место для его расположения на диске. При создании нового проекта автоматически запускается новое решение. Проект можно добавить к существующему решению с помощью обозревателя решений.

Для каждого созданного проекта в обозревателе решений отображаются папки **Connections** (Соединения), **Queries** (Запросы) и **Miscellaneous** (Разное). Чтобы открыть новое окно редактора запросов для данного проекта, щелкните правой кнопкой его папку **Queries** (Запросы) и в контекстном меню выберите пункт **New Query** (Создать запрос).

## Отладка SQL Server

SQL Server, начиная с версии SQL Server 2008, оснащен встроенным отладчиком кода. Чтобы начать сеанс отладки, выберите в главном меню среды SQL Server Management Studio следующую последовательность команды **Debug | Start Debugging** (Отладка | Начать отладку). Мы рассмотрим работу отладчика на примере с использованием пакета из *главы 8* (см. *пример 8.1*). *Пакетом* называется последовательность инструкций SQL и процедурных расширений, составляющих логическое целое, отправляемая компоненту Database Engine для выполнения всех содержащихся в ней инструкций. На рис. 3.13 показан пакет, который подсчитывает количество сотрудников, работающих над проектом p1. Если это количество равно 4 или больше, то выводится соответствующее сообщение. В противном случае выводятся имена и фамилии сотрудников.

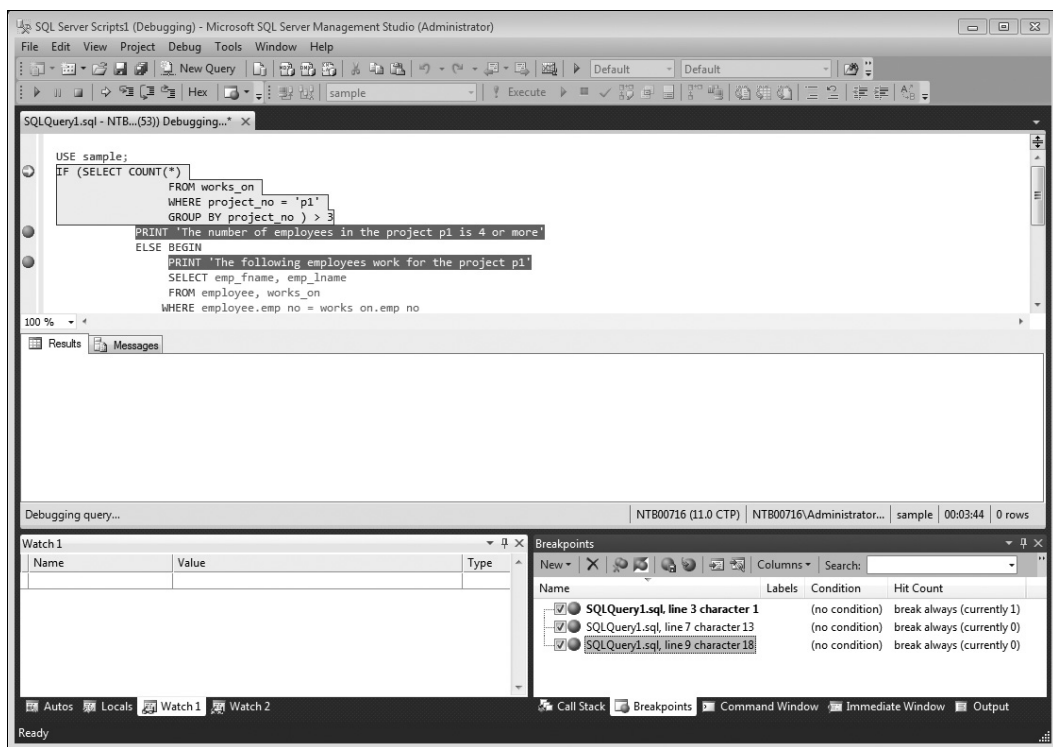


Рис. 3.13. Отладка кода SQL

Чтобы остановить выполнение пакета на определенной инструкции, можно установить точки останова, как это показано на рис. 3.13. Для этого нужно щелкнуть слева от строки, на которой нужно остановиться. В начале отладки выполнение останавливается на первой линии кода, которая отмечается желтой стрелкой. Чтобы продолжить выполнение и отладку, выберите следующую последовательность команд из меню **Debug | Continue** (Отладка | Продолжить). Выполнение инструкций пакета продолжится до первой точки останова, и желтая стрелка остановится на этой точке.

Информация, связанная с процессом отладки, отображается в двух панелях внизу окна редактора запросов. Информация о разных типах информации об отладке сгруппирована в этих панелях на нескольких вкладках. Левая панель содержит вкладку **Autos** (Автоматические), **Locals** (Локальные) и до пяти вкладок **Watch** (Видимые). Правая панель содержит вкладки **Call Stack** (Стек вызовов), **Threads** (Потоки), **Breakpoints** (Точки останова), **Command Window** (Окно команд), **Immediate Window** (Окно интерпретации) и **Output** (Вывод). На вкладке **Locals** (Локальные) отображаются значения переменных, на вкладке **Call Stack** (Стек вызовов) — значения стека вызовов, а на вкладке **Breakpoints** (Точки останова) — информация о точках останова.

Чтобы завершить процесс отладки, выполните последовательность команд из главного меню **Debug | Stop Debugging** (Отладка | Остановить отладку) или нажмите синюю кнопку на панели инструментов отладчика.



В SQL Server 2012 функциональность встроенного в SQL Server Management Studio отладчика расширена несколькими новыми возможностями. Теперь в нем можно выполнять ряд следующих операций.

- ◆ *Указывать условие точки останова.* Условие точки останова — это SQL-выражение, вычисленное значение которого определяет, будет ли выполнение кода остановлено в данной точке или нет. Чтобы указать условие точки останова, щелкните правой кнопкой красный значок требуемой точки и в контекстном меню выберите пункт **Condition** (Условие). Откроется диалоговое окно **Breakpoint Condition** (Условие для точки останова), в котором нужно ввести необходимое логическое выражение. Кроме этого, если нужно остановить выполнение, в случае если выражение верно, то следует установить переключатель **Is True** (Верно). Если же выполнение нужно остановить, если выражение изменилось, то нужно установить переключатель **Has Changed** (Изменилось).
- ◆ *Указать число попаданий в точку останова.* Число попаданий — это условие останова выполнения в данной точке в зависимости от количества раз, когда была достигнута эта точка останова в процессе выполнения. При достижении указанного числа прохождений и любого другого условия, указанного для данной точки останова, отладчик выполняет указанное действие. Условие прерывания выполнения на основе числа попаданий может быть одним из следующих:
  - безусловное (действие по умолчанию) (**Break always**);
  - если число попаданий равно указанному значению (**Break when the hit count equals a specified value**);
  - если число попаданий кратно указанному значению (**Break when the hit count equals a multiple of a specified value**);
  - если число попаданий равно или больше указанного значения (**Break when the hit count is greater or equal to a specified value**).

Чтобы задать число попаданий в процессе отладки, щелкните правой кнопкой значок требуемой точки останова на вкладке **Breakpoints** (Точки останова), в контекстном меню выберите пункт **Hit Count** (Число попаданий), затем в открывшемся диалоговом окне **Breakpoint Hit Count** (Число попаданий в точку останова) выберите одно из условий из приведенного ранее списка. Для опций, требующих значение, введите его в текстовое поле справа от раскрывающегося списка условий. Чтобы сохранить указанные условия, нажмите кнопку **OK**.

- ◆ *Указывать фильтр точки останова.* Фильтр точки останова ограничивает работу останова только на указанных компьютерах, процессах или потоках. Чтобы установить фильтр точки останова, щелкните правой кнопкой требуемую точку и в контекстном меню выберите пункт **Filter** (Фильтр). Затем в открывшемся диалоговом окне **Breakpoint Filters** (Фильтр точки останова) укажите ресурсы, которыми нужно ограничить выполнение данной точки останова. Чтобы сохранить указанные условия, нажмите кнопку **OK**.
- ◆ *Указывать действие в точке останова.* Условие **When Hit** (При попадании) (в точку останова) указывает действие, которое нужно выполнить, когда выполнение пакета попадает в данную точку останова. По умолчанию, когда удовле-

творяются как условие количества попаданий, так и условие останова, тогда выполнение прерывается. Альтернативно можно вывести заранее указанное сообщение.

Чтобы указать действие при попадании в точку останова, щелкните правой кнопкой красный значок требуемой точки и выберите в контекстном меню пункт **When Hit** (При попадании). В открывшемся диалоговом окне **When Breakpoint is Hit** (При попадании в точку останова) выберите требуемое действие. Чтобы сохранить указанные условия, нажмите кнопку **ОК**.

- ◆ *Использовать окно быстрой проверки Quick Watch.* В окне **QuickWatch** (Быстрая проверка) можно просмотреть значение выражения Transact-SQL, а потом сохранить это выражение в окне просмотра значений **Watch** (Просмотр значений). Чтобы открыть окно **Quick Watch** (Быстрая проверка), в меню **Debug** (Отладка) выберите пункт **Quick Watch** (Быстрая проверка). Выражение в этом окне можно или выбрать из раскрывающегося списка **Expression** (Выражение), или ввести его в это поле.
- ◆ *Использовать всплывающую подсказку Quick Info.* При наведении указателя мыши на идентификатор кода средство **Quick Info** (Краткие сведения) отображает его объявление во всплывающем окне.

## Резюме

В этой главе мы рассмотрели самый важный инструмент сервера SQL Server: среду управления SQL Server Management Studio. Среда SQL Server Management Studio очень полезна как для конечных пользователей, так и для администраторов. С ее помощью можно осуществлять многие функции администрирования. Эти функции были только слегка затронуты в этой главе, но рассматриваются более подробно далее в книге. В этой главе мы познакомились с наиболее важными для конечного пользователя функциями среды SQL Server Management Studio, такими как создание баз данных и таблиц.

Среда управления SQL Server Management Studio содержит, среди прочих, следующие инструменты:

- ◆ *Зарегистрированные серверы (Registered Servers)* — используется для регистрации экземпляров SQL Server и подключения к ним;
- ◆ *Обозреватель объектов (Object Explorer)* — содержит в виде дерева представление всех объектов баз данных сервера;
- ◆ *Редактор запросов (Query Editory)* — позволяет конечным пользователям создавать, выполнять и сохранять инструкции Transact-SQL. Кроме этого, данный инструмент предоставляет возможность для анализа запросов, отображая план выполнения;
- ◆ *Обозреватель решений (Solution Explorer)* — используется для создания решений; причем решение может быть связано ни с одним, с одним или с несколькими проектами;
- ◆ *Отладчик (Debugger)* — используется для отладки SQL-кода.