

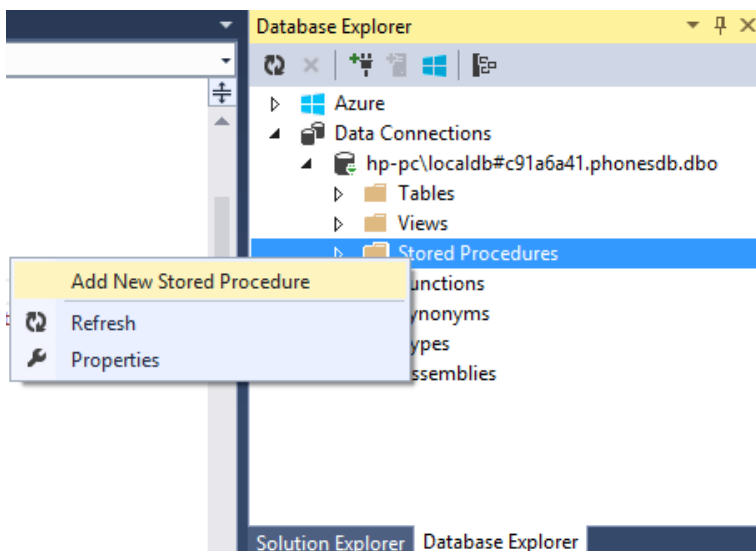


## Хранимые процедуры

Последнее обновление: 31.10.2015



Работа с хранимыми процедурами во многом аналогична работе с функциями базы данных. Итак, возьмем бд из прошлой темы и добавим в нее новую процедуру. Для этого откроем подключение к базе данных в окне Database Explorer и найдем в его структуре узел **Stored Procedures** (Хранимые процедуры). Нажмем на этот узел правой кнопкой мыши и в появившемся меню выберем **Add New Stored Procedure**:



После этого Visual Studio сгенерирует код процедуры по умолчанию:

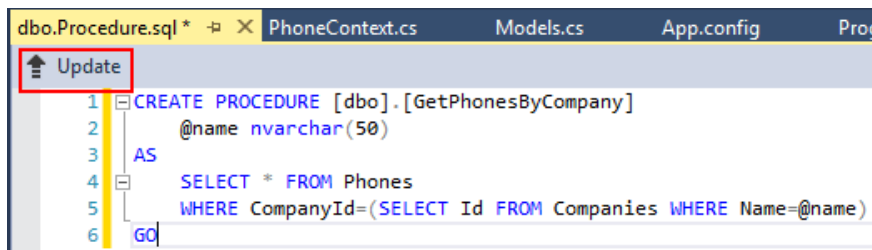
```
CREATE PROCEDURE [dbo].[Procedure]
    @param1 int = 0,
    @param2 int
AS
    SELECT @param1, @param2
RETURN 0
```

Изменим ее следующим образом:

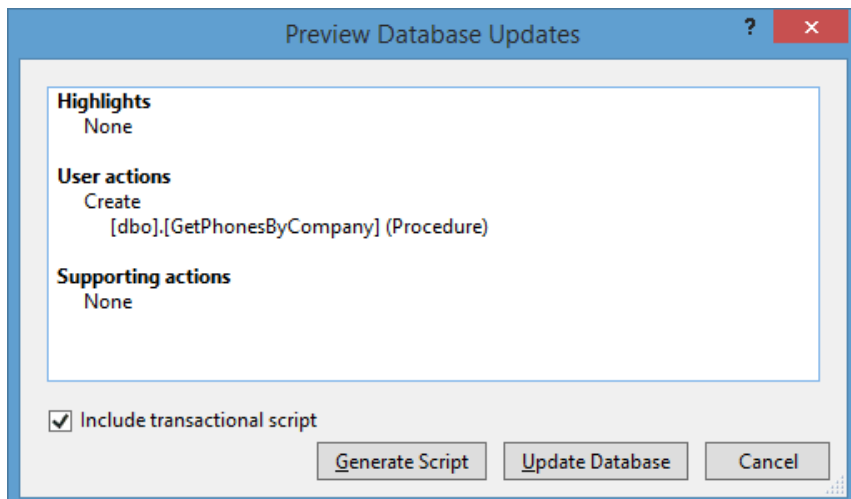
```
CREATE PROCEDURE [dbo].[GetPhonesByCompany]
    @name nvarchar(50)
AS
    SELECT * FROM Phones
    WHERE CompanyId=(SELECT Id FROM Companies WHERE Name=@name)
GO
```

Данная процедура ищет все строки, где значение столбца название компании равно строке, переданной через параметр @name.

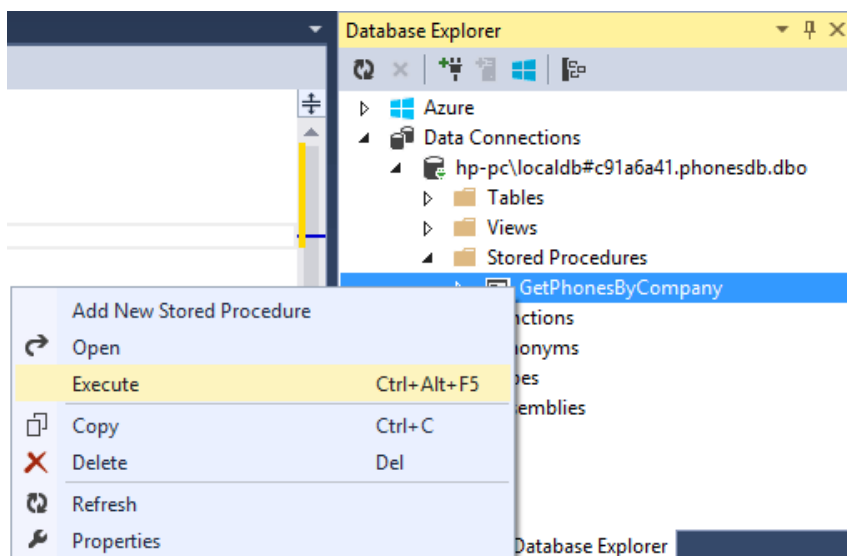
Далее нажмем на кнопку Update для добавления хранимой процедуры:



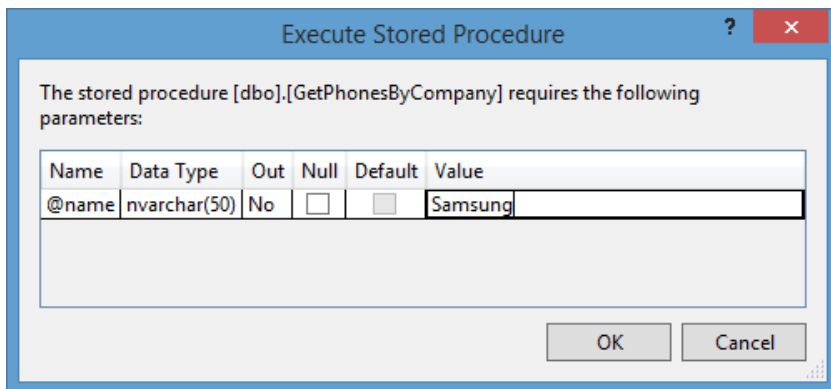
И затем в появившемся окошке нажимаем кнопку Update Database:



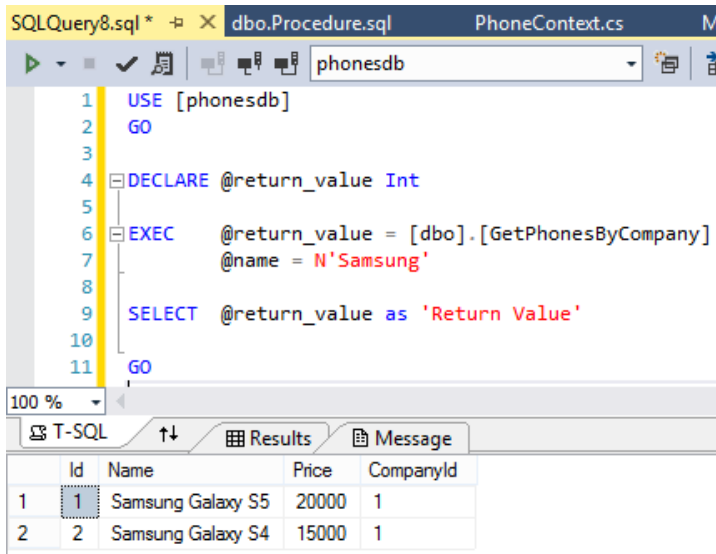
После этого в узле **Stored Procedures** появится новая хранимая процедура. Перед ее использованием мы можем проверить ее, чтобы убедиться, что она работает как надо и получает нужные данные. Для этого нажмем на нее правой кнопкой мыши и выберем в меню **Execute**:



После этого нам откроется окно настройки входного параметра. В поле Value введем какое-нибудь имя, по которому будем осуществлять поиск и нажмем ОК:



И Visual Studio автоматически генерирует и запускает скрипт на языке SQL с вызовом процедуры и передачи ей параметра:



Теперь обратимся к процедуре в коде C#:

```
using (PhoneContext db = new PhoneContext())
{
    System.Data.SqlClient.SqlParameter param = new System.Data.SqlClient.SqlParameter("@name", "Samsung");
    var phones = db.Database.SqlQuery<Phone>("GetPhonesByCompany @name", param);
    foreach (var p in phones)
        Console.WriteLine("{0} - {1}", p.Name, p.Price);
}
```

Параметр в методе `SqlQuery` принимает название процедуры, после которого идет перечисление параметров: `GetPhonesByCompany @name`

И так как процедура возвращает строки из таблицы `Phones`, то метод `SqlQuery` мы можем типизировать классом `Phone`: `db.Database.SqlQuery<Phone>(...)`

[Назад](#) [Содержание](#) [Вперед](#)



---

[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.