Contents

[Redgate and Visual Studio 2017 1](#_Toc477531254)

[The demo video can be watched here . 1](#_Toc477531255)

[The demo source code can be downloaded from here . 1](#_Toc477531256)

[Demo Overview 1](#_Toc477531257)

[Rough Outline 1](#_Toc477531258)

[Key Takeaways 2](#_Toc477531259)

[VS Configuration 2](#_Toc477531260)

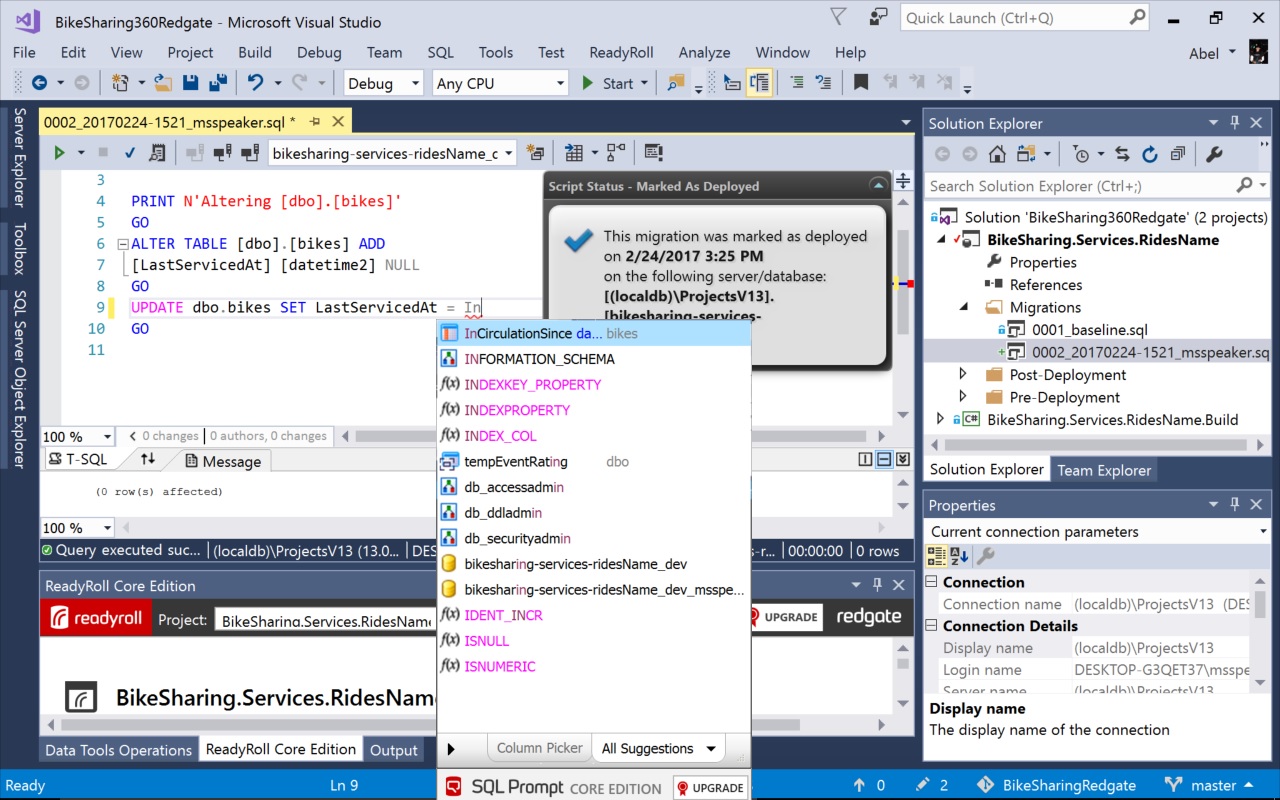
[Task 1: Reset demo repository 2](#_Toc477531261)

[Task 2: Configure Visual Studio 2](#_Toc477531262)

[Task 3: Reset databases 3](#_Toc477531263)

[Demo 4](#_Toc477531264)

**Redgate and Visual Studio 2017**

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/ReadyRoll.jpg)

The demo video can be watched [here](https://channel9.msdn.com/Events/Visual-Studio/Visual-Studio-2017-Launch/100#time=1h54m38s) .

The demo source code can be downloaded from [here](https://github.com/abelsquidhead/VS2017VSTSKeynoteDemo) .

**Demo Overview**

In this demo, Donovan Brown, will show:

**High-level flow**

1. Redgate Data Tools now available for FREE to all VSE users (in the VSE installer)
2. Don’t let your database slow down your releases, develop and deploy your application and your database side-by-side in existing MS tools (VS, TFS/VSTS) using Redgate ReadyRoll
3. Migrations shifts the creation of the database deployment script as far left as possible so that it can be reviewed and tested early (more control over deployments)
4. SQL Prompt makes writing SQL code easier and smarter

**Rough Outline**

Three of Redgate’s tools are now available to install in Visual Studio Enterprise – Redgate ReadyRoll Core, Redgate SQL Prompt Core, and Redgate SQL Search.

Now that BikeSharing360 has been live for a while, we want to start thinking about maintenance of our bikes – Redgate ReadyRoll makes it easy to make and deploy these changes. We’ll add a column to the bikes table that lists the last date that a bike was serviced. We’ll use VSTS to add these changes to source control, showing how easy it is to make these deployments and show a preview of what changes will be made.

**Key Takeaways**

1. Don’t let your database slow down your releases, develop and deploy your application and your database side-by-side in existing MS tools (VS, TFS/VSTS) using Redgate ReadyRoll.
2. ReadyRoll is an alternative to SSDT Database Projects that allows developers to have more control over the end database deployment script. ReadyRoll is a migrations-driven approach which shifts the creation of the database deployment script left so that it can be source controlled and tested early for fast feedback. This gives developers more control over exactly how database changes will be deployed to the different environments.
3. SQL Prompt makes writing SQL code easier and smarter

**VS Configuration**

Install Visual Studio 2017 RC by selecting the “Data storage and processing workload” and making sure that all three Redgate tools (Redgate ReadyRoll, Redgate SQL Prompt, and Redgate SQL Search) are checked.

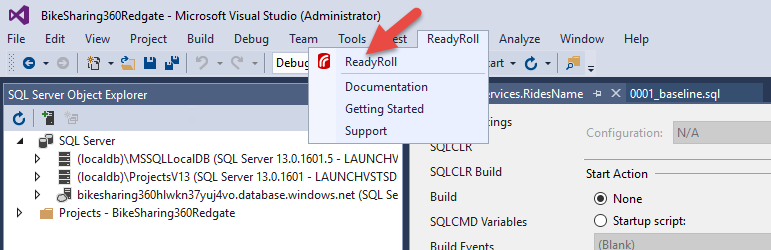
Open Visual Studio and sign in.

**Task 1: Reset demo repository**

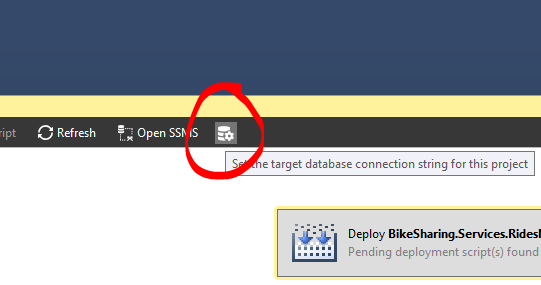
1. Open the command prompt or PowerShell
2. git clone <https://github.com/abelsquidhead/VS2017VSTSKeynoteDemo>
3. git push origin +5a5d6:master (please use the hash in git before you made your changes)
4. git reset --hard origin/master

**Task 2: Configure Visual Studio**

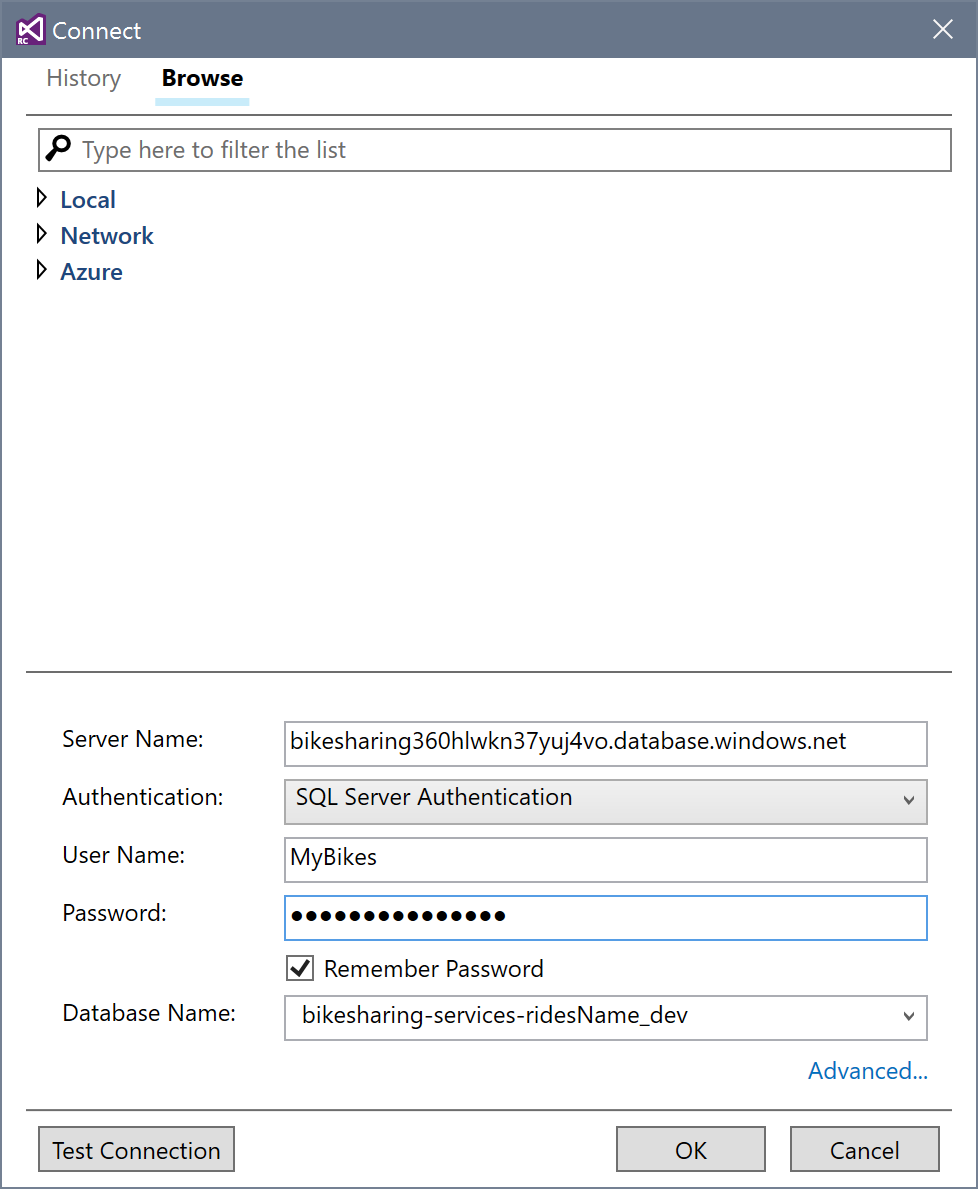
1. Open **BikeSharing360Redgate.sln** from the folder that you cloned the project into.
2. Open the ReadyRoll window. If it isn’t already there, click **ReadyRoll** | **ReadyRoll**

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image1.png)

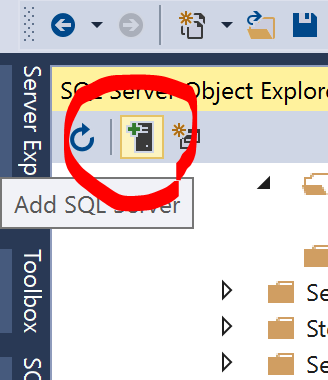
1. In the ReadyRoll window at the bottom click the button labeled **Set the target database connection string for this project**

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image3.png)

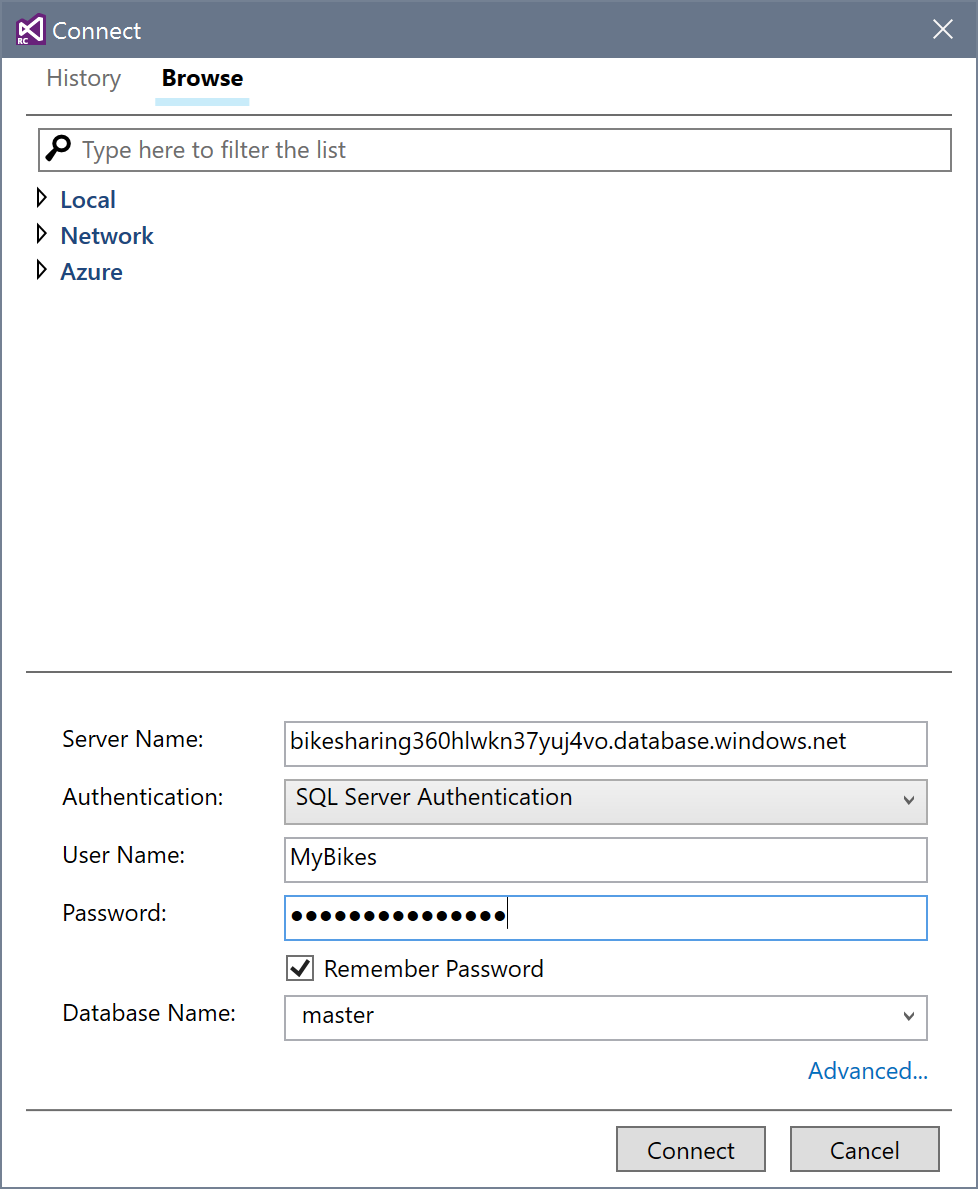
1. Use the following settings for setting the target database connection:
   * **Server name**: git pus
   * **Authentication**: SQL Server Authentication
   * **User name**: MyBikes
   * **Password**: BikesShare360@1
   * **Database name**: bikesharing-services-ridesName\_dev

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image4.PNG)

1. Click **Test Connection**.
2. Click **OK**.
3. Open the SQL Server Object Explorer
4. Click the button labeled **Add SQL Server**.

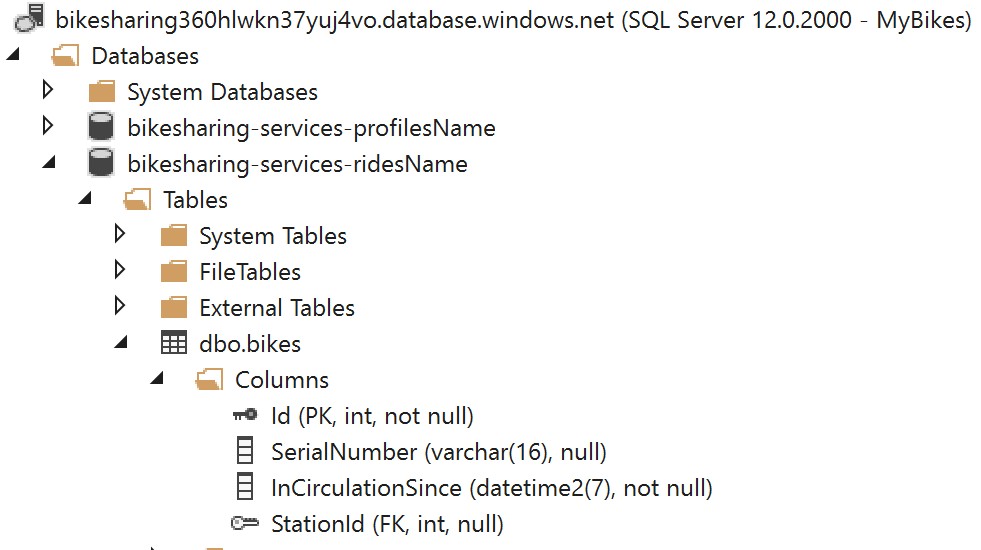
[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image5.PNG)

1. Enter the following:
   * **Server name**: bikesharing360hlwkn37yuj4vo.database.windows.net
   * **Authentication**: SQL Server Authentication
   * **User name**: MyBikes
   * **Password**: BikesShare360@1
   * **Database name**: master

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image6.PNG)

**Task 3: Reset databases**

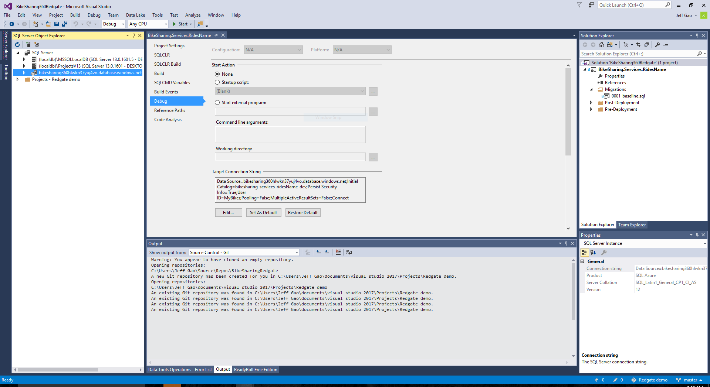
1. Open the SQL Server Object Explorer in Visual Studio
2. Expand the following:
   * bikesharing360hlwkn37yuj4vo.database.windows.net
   * Databases
   * bikesharing-services-ridesName
   * Tables
   * dbo.bikes
   * Columns

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image7.PNG)

1. If there are any columns other than these ones listed **(Id, SerialNumber, InCirculationSince, StationId)**, right click the column and click **Delete**.
2. Repeat these steps for **bikesharing-services-ridesName\_dev**.

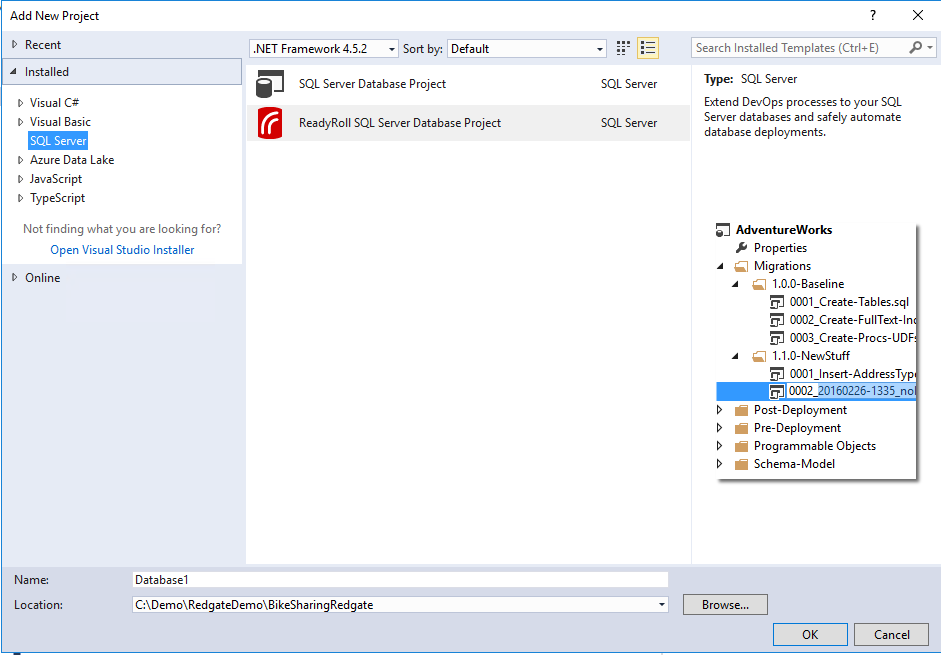
**Demo**

1. Open the project **BikeSharing.Services.RidesName** in your Visual Studio 2017

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image8.png)

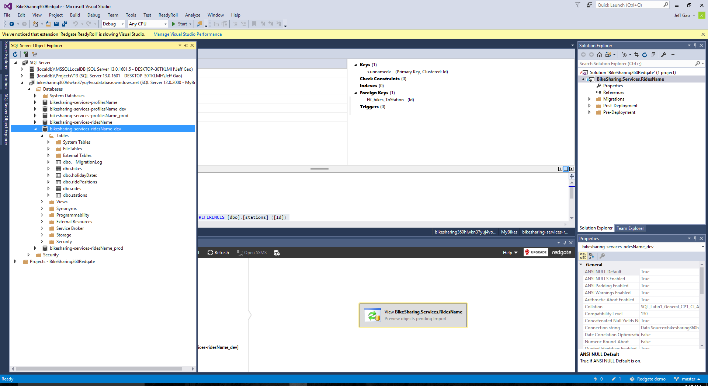
**Talking Point**: Managing your database changes and incorporating them into your DevOps pipeline has always been difficult. Keeping track of what scripts need to be run, getting the database states just right is a pain and very error prone. However, there are some fantastic tools from Redgate which help us manage our database changes as well as creating a fully automated CI/CD Pipeline. With each copy of Visual Studio Enterprise, you get Ready Roll, SQL Prompt, and SQL Search for free. Now, with VSE and the included Redgate tools, you can check in and out your database schema right alongside your source code. You can also easily create your CI/CD pipeline which will seamlessly deploy those changes to the database’s in all your environments.

1. Right click solution and **add | new | project**

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image9.png)

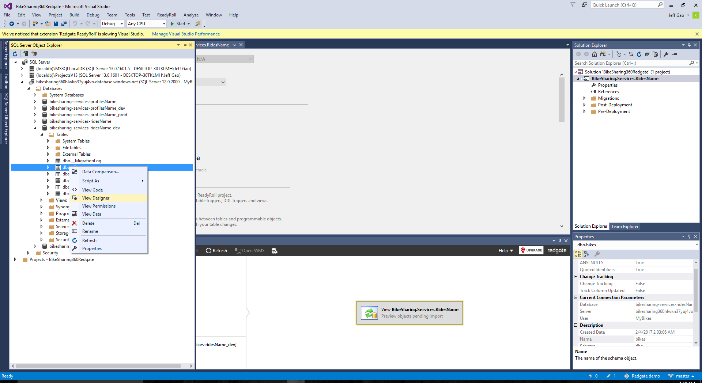
**Talking Point**: To create a database project from scratch is super easy, just right click your solution, add new project, click on the SQL server node in the tree and select ReadyRoll SQL Server Database project

1. Expand **bikesharing-services-ridesName\_dev** database in SQL Server Object Explorer. Expand the Tables folder.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image10.png)

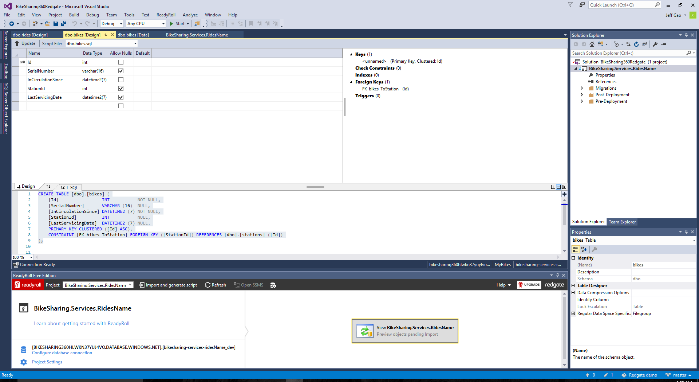
**Talking Point**: We’ve already created one for this demo and it is already connected to our dev instance.

1. Right click **dbo.bikes** and select **View Designer**.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image11.png)

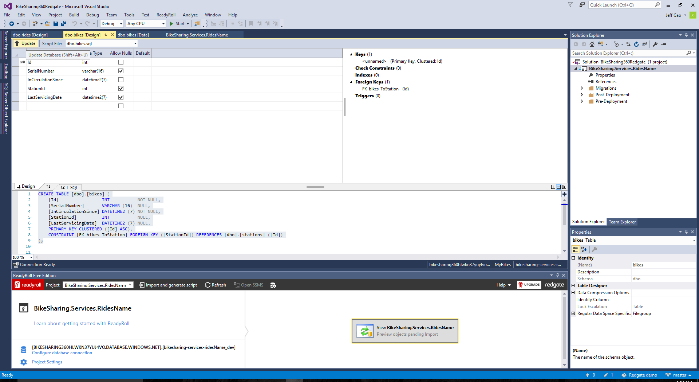
**Talking Point**: If I need to change the schema of my database, all I need to do is just change the schema in SQL Server Object Explorer or SSMS or whatever tool of choice and make my change to the DB schema. ReadyRoll detects those changes you made to your dev database and will add the correct migration script to your project.

1. In the Designer window, add a **new SQL column**. Set its name to **LastServicedAt**, its data type to **datetime2(7)**, and make sure the checkbox under **allow nulls** is checked.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image12.png)

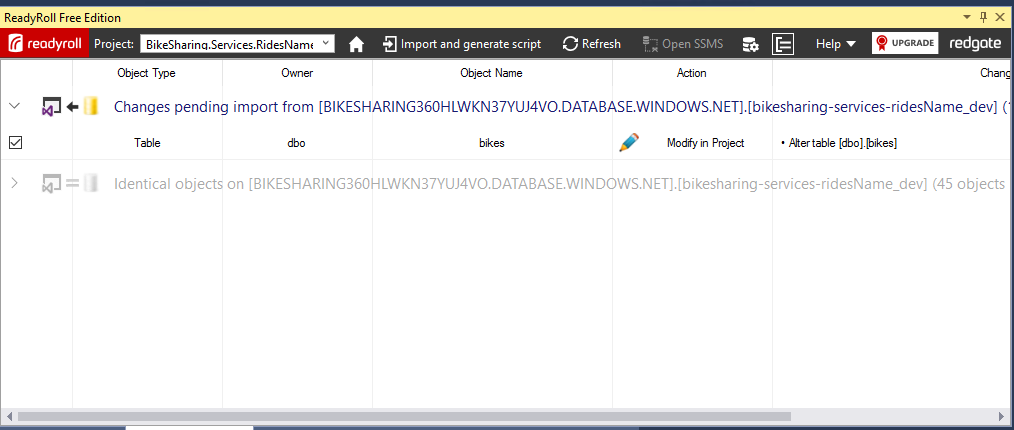
**Talking Point**: In this project, I’m going to need to add a new column to the bikes table. I need to add a LastServicedAt column.

1. Click **update** in the top left corner.

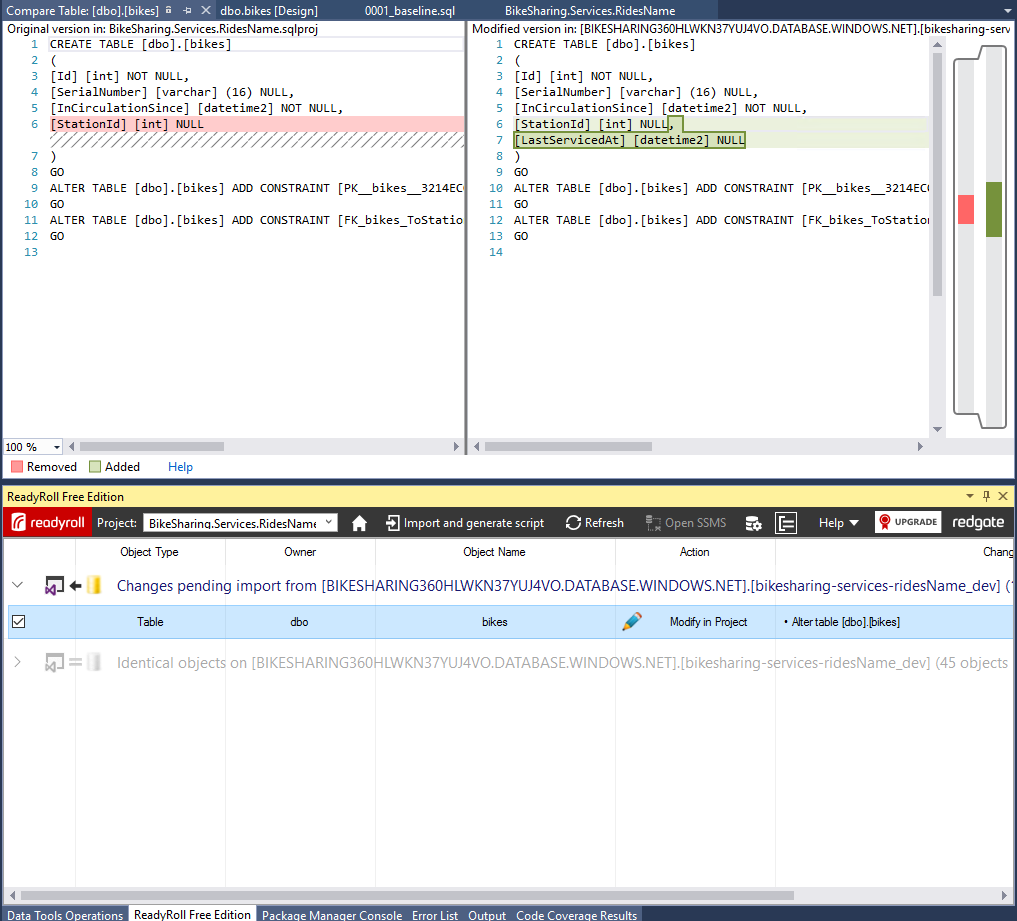
[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image13.png)

**Talking Point**: When I’m done modifying my database, I go ahead and click the update and it will update my dev database.

1. Open the ReadyRoll window by clicking the **ReadyRoll Core Edition** tab at the bottom, then click the **Refresh button**.

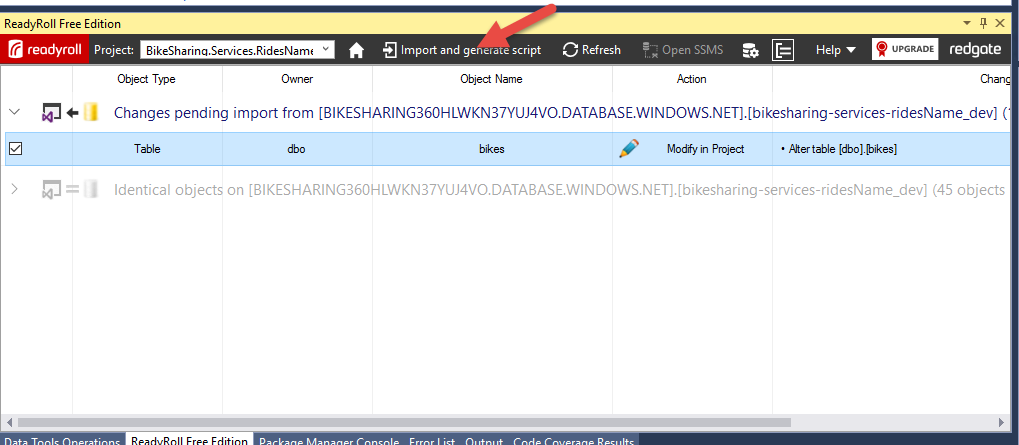
[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image14.png)

**Talking Point**: Next, I go into the ReadyRoll window and click refresh. This will do a schema compare between my dev database and what ReadyRoll thinks my database is currently, and will automatically generate the migration script.

1. Double click on the row. A tab will open displaying the changes between our old and new schema.  
   [](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image15.png)

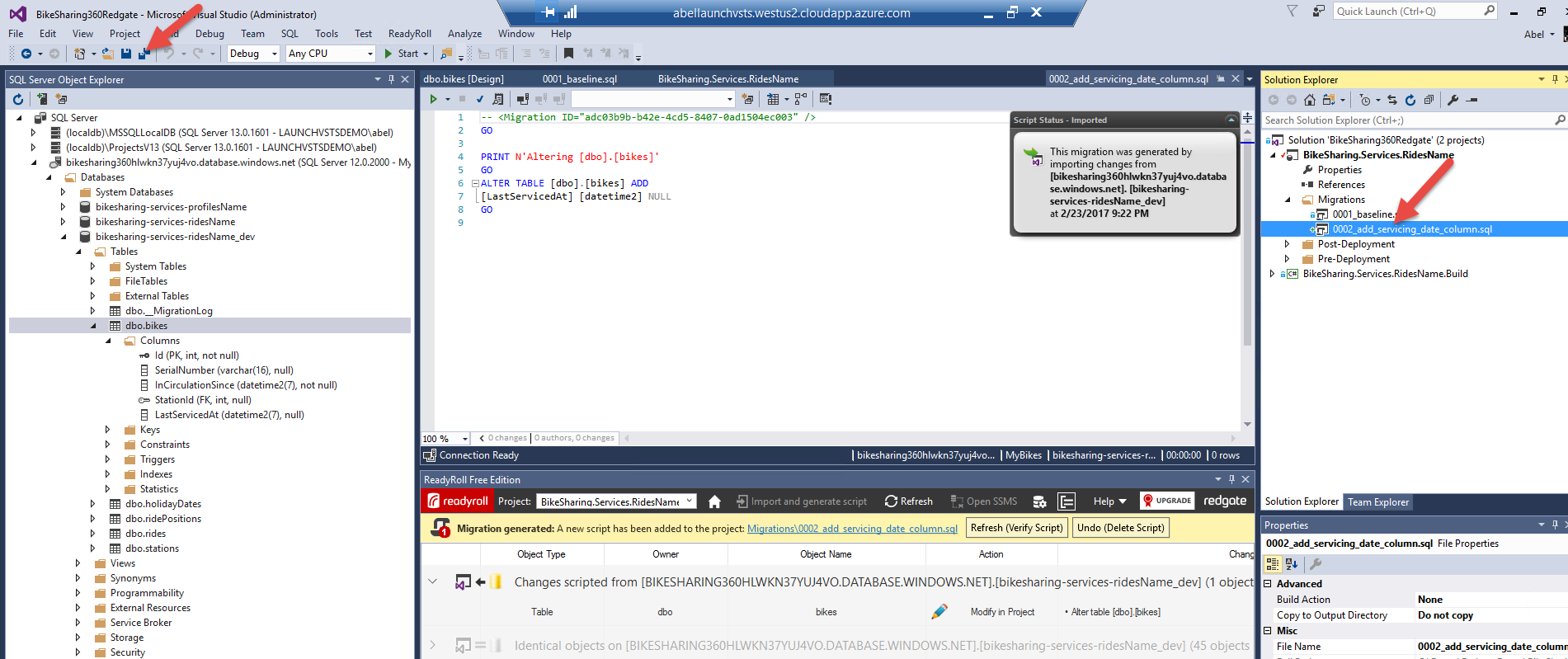
**Talking Point**: To see exactly what it found, we can go ahead and double click on the changes, this will bring up a diff of the changes.

1. Click **generate script(s)** for all database changes that are pending import.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image16.png)

**Talking Point**: To bring this migration script into the solution, I’ll go and click the Import and generate script button.

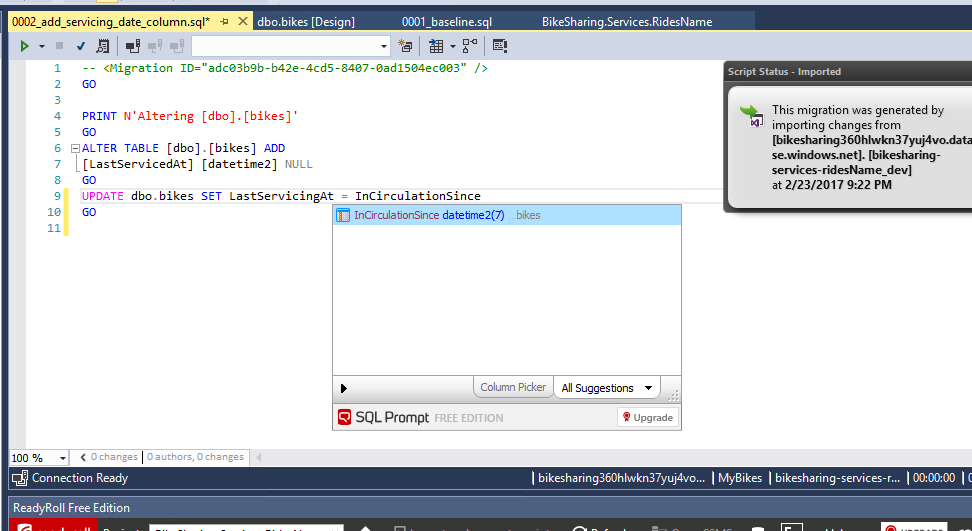
1. Rename the migration script to **0002\_add\_servicing\_date\_column.sql”**. After renaming, make sure you click the save all button so the renamed file is saved and the project reference is saved.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image17.png)

**Talking Point**: Lets change the name of the imported script so it’s easier to read.

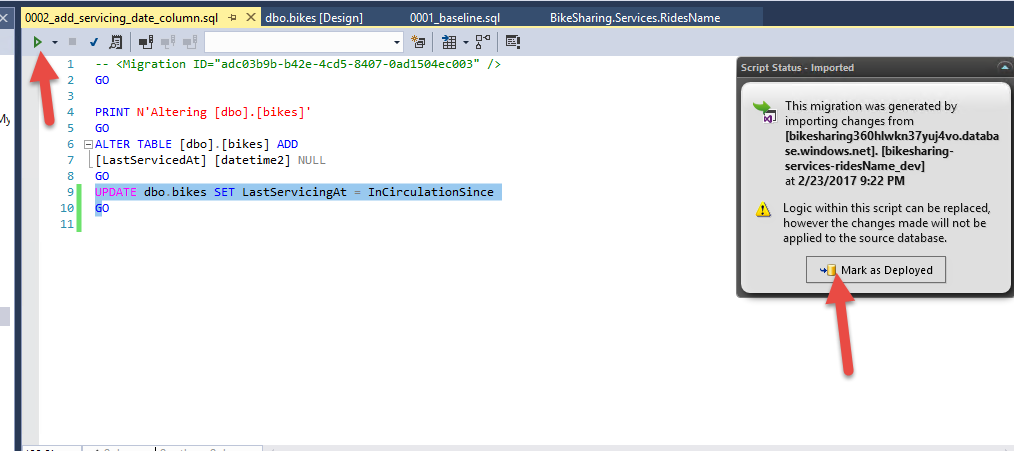
1. Double click the migration script **(0002\_add\_servicing\_date\_column.sql)** to open it. In the new SQL query file, type the below code snippet at the end (after the last “GO”)
2. GO
3. UPDATE dbo.bikes SET LastServicingAt = InCirculationSince
4. GO

After typing this make sure you save the file.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image18.png)

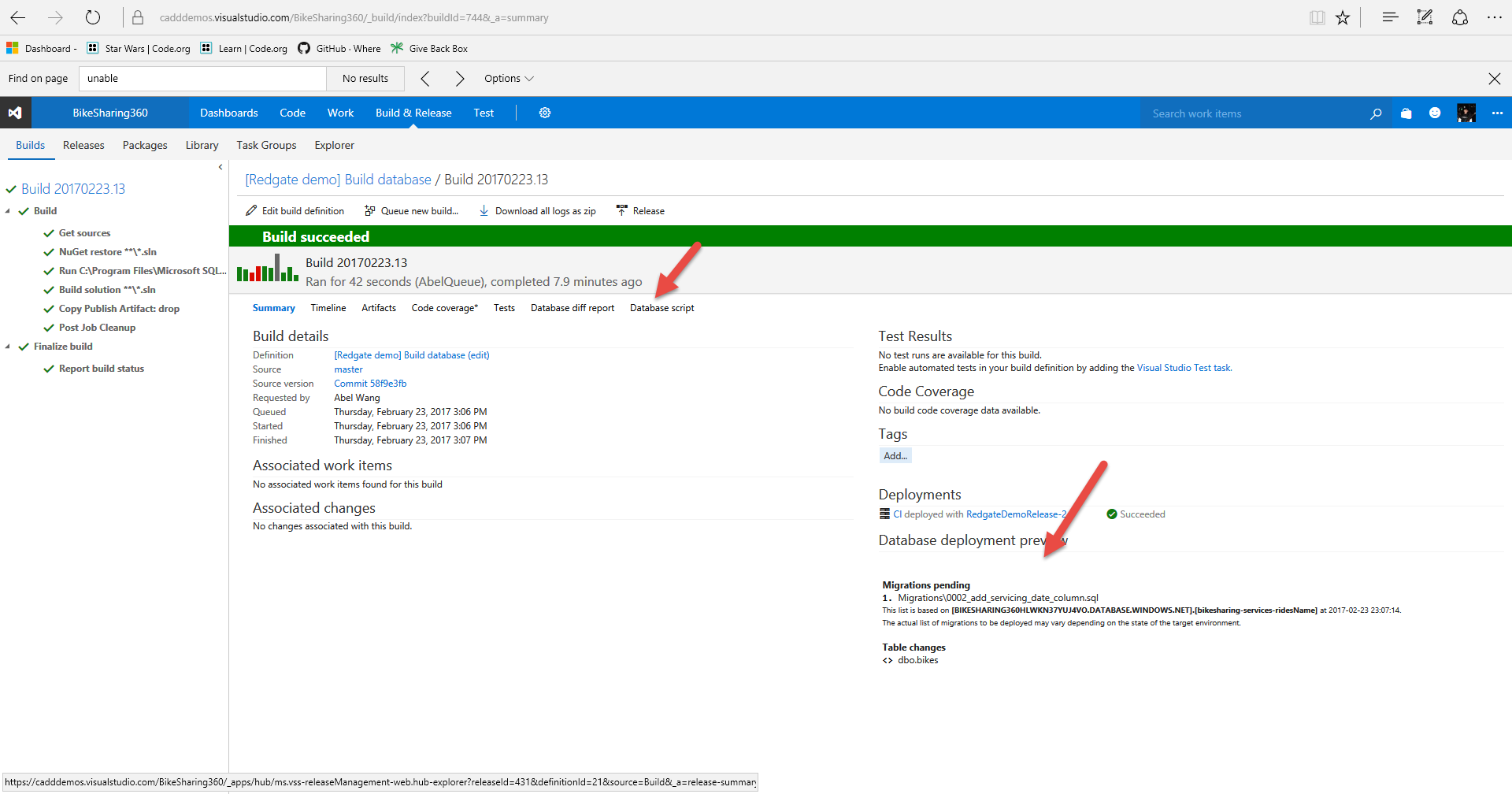
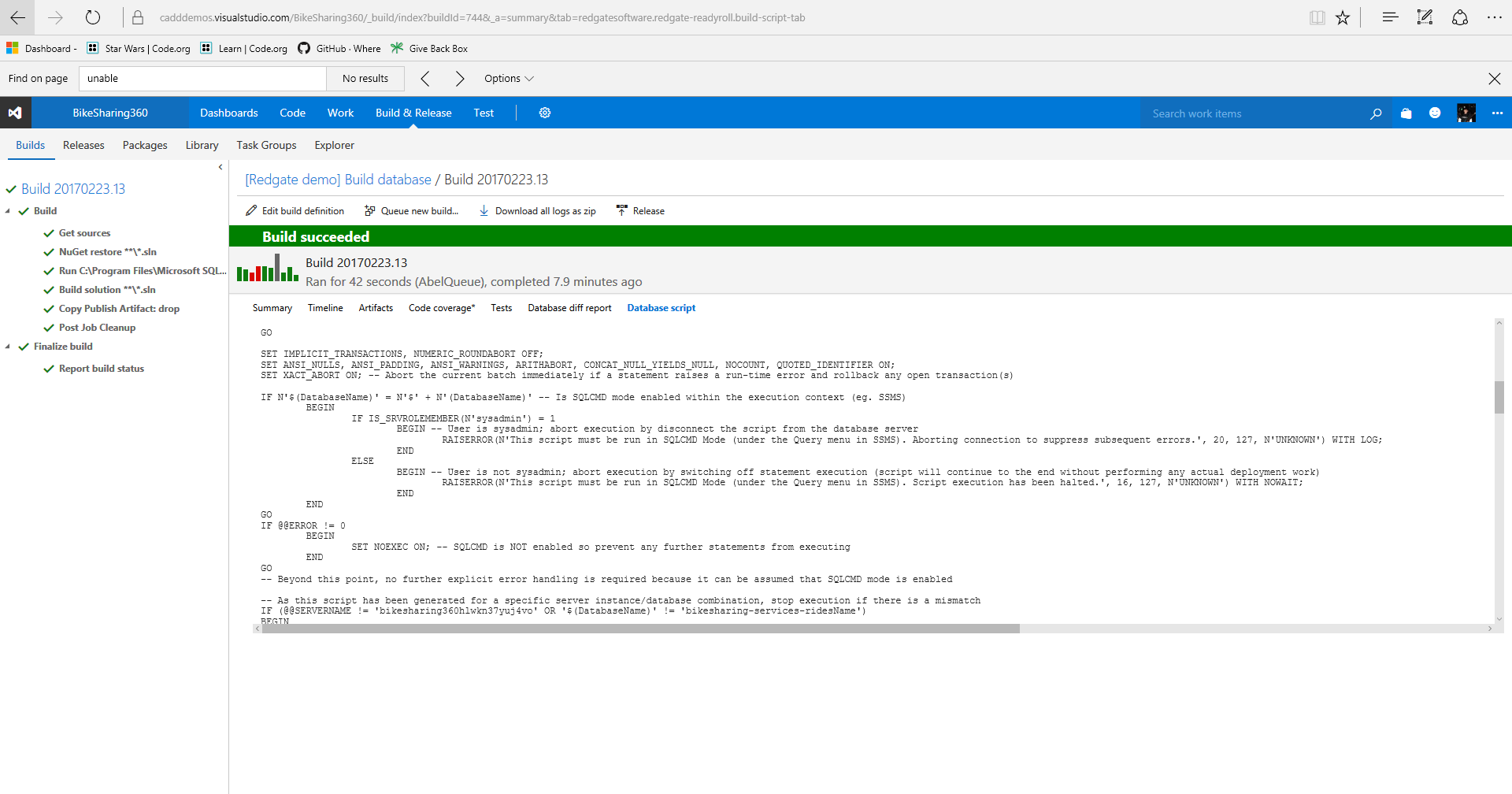
**Talking Point**: It’s super easy for us to prepopulate this column with data too. Just double click the migration script and modify it. Notice how SQL Prompt Core offers suggestions and gives quality of life improvements for writing SQL code?

1. Select these new lines and click the **Execute** button **(or press ctrl-shift-e)** to run this code on the dev database. Click **Mark as Deployed** in the **Script Status** window.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image19.png)

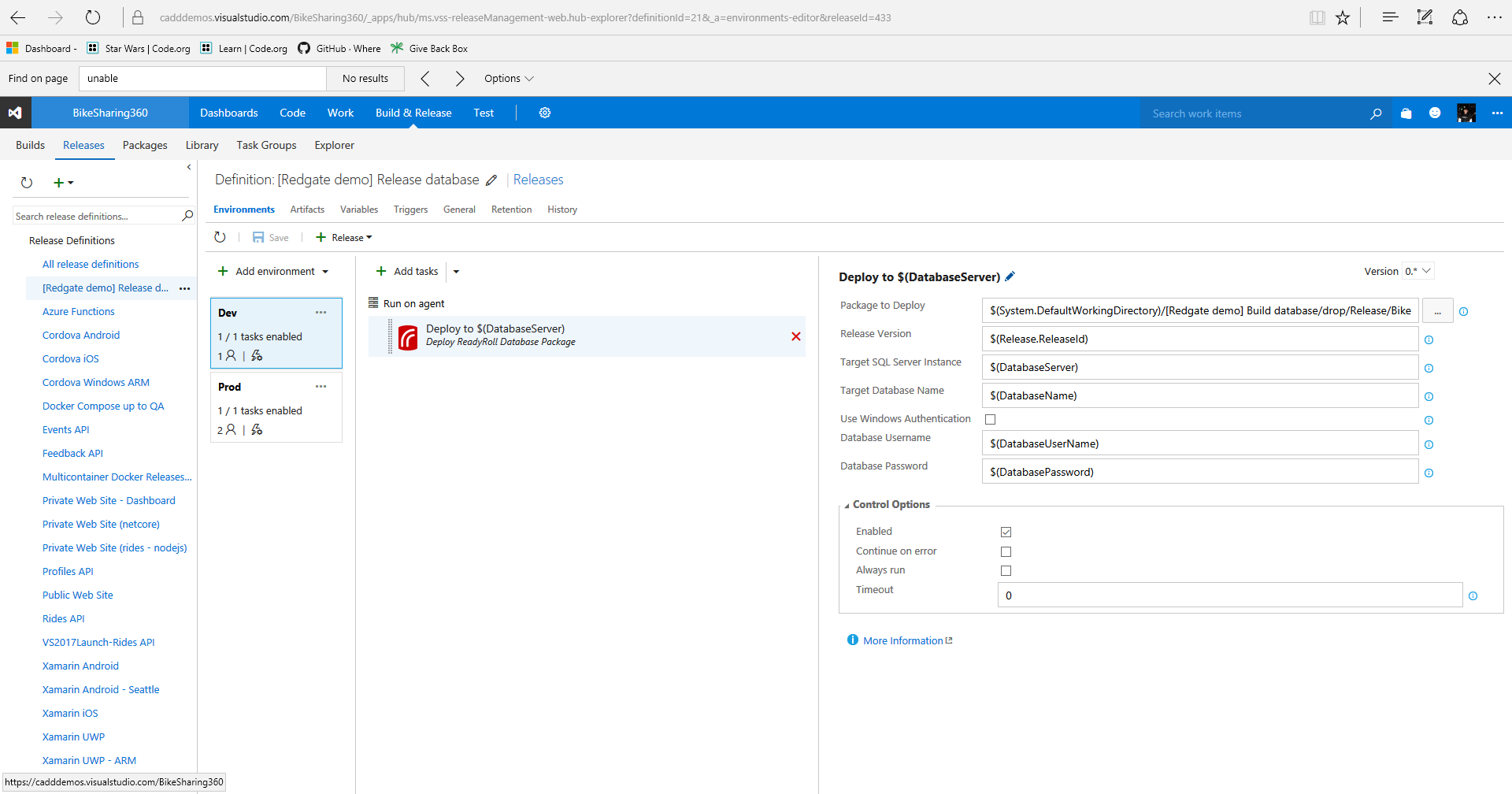
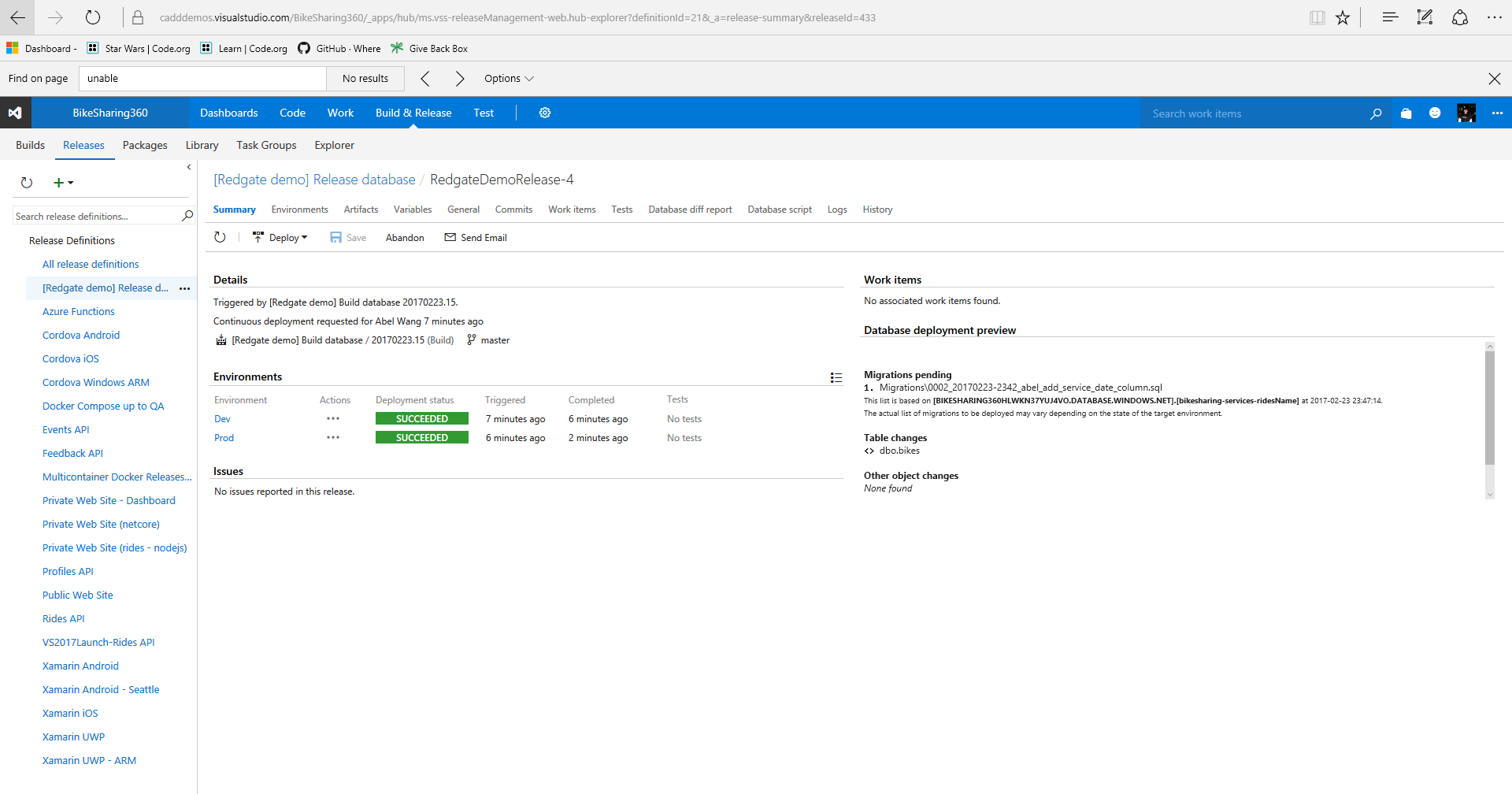
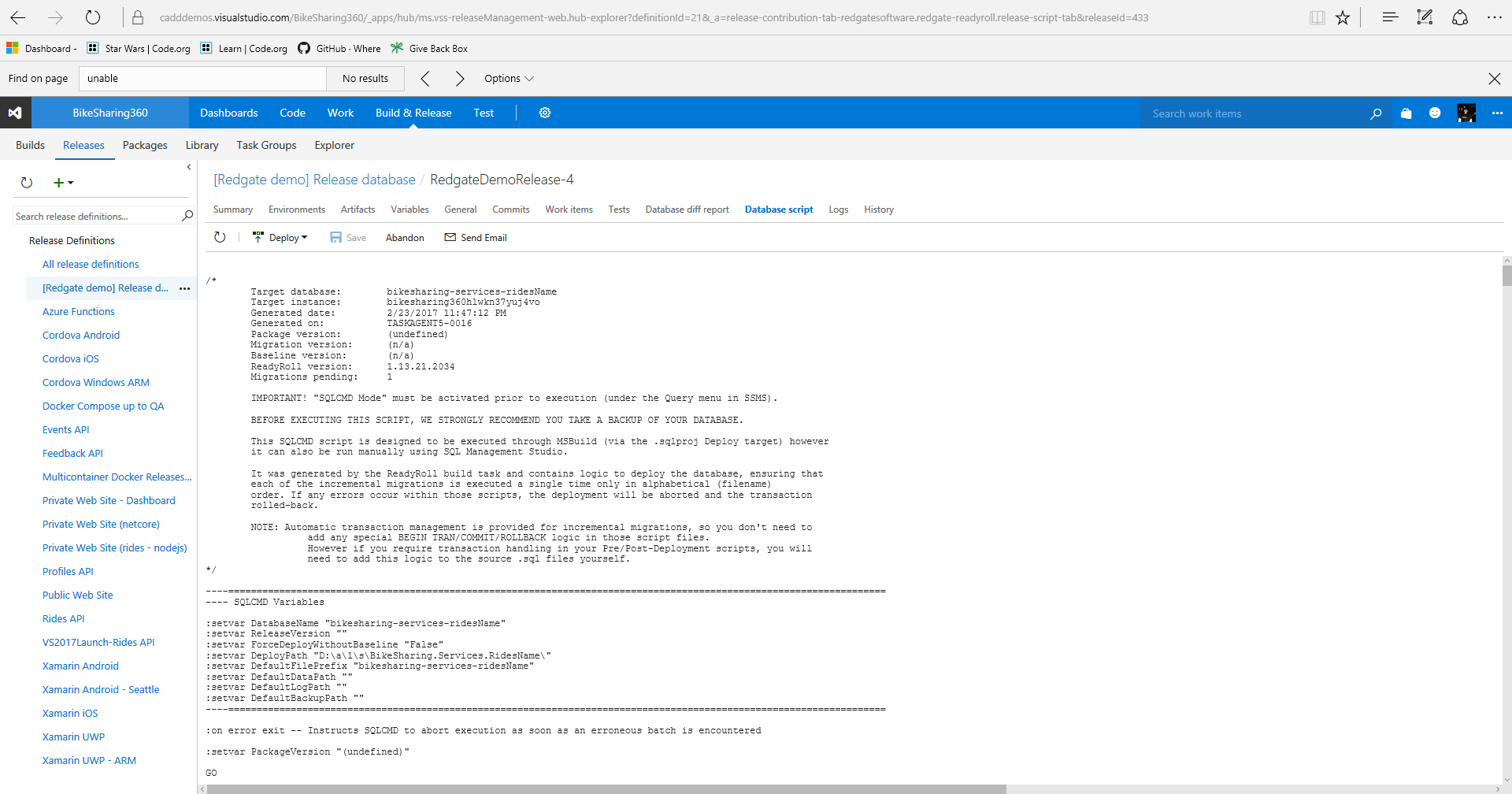
**Talking Point**: We’ll go ahead and run this script to populate the new column with data in our dev database. Since we haven’t serviced our bikes yet, LastServicing = InCirculationSince. ReadyRoll also makes it easy to add NOT NULL columns without a DEFAULT.

1. Check code back into **VSTS** and go to VSTS portal and check out the build in progress.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image20.png)[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image21.png)

**Talking Point**: So now that we made those changes, let’s check in our code. This will kick off our build. As you can see, the build is just a simple straight forward build. The build takes a while so let’s again take a look at a build that has already gone through. Notice in our build report, it lists the tables that have changed. Also, if we click on Database script, we can actually see the migration scripts that get run for the deployments.

1. Go to **release**, show the task. Click on old release, show the database deployment preview section and also the Database script tab.

[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image22.png)[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image23.png)[](https://github.com/Microsoft/VisualStudio2017Launch_KeynoteDemos/blob/master/Redgate%20and%20Visual%20Studio%202017/media/image24.png)

**Talking Point**: Now our build is linked to RM, so when we get a successful build, it kicks off our CD pipeline where we can easily deploy the database changes using prebuild release ReadyRoll Task. Once again, the release takes a while so let’s look at a prior release. We can see exactly what tables were changed, what migrations were ran. We can look at the Data script tab and see the exact script used as well.