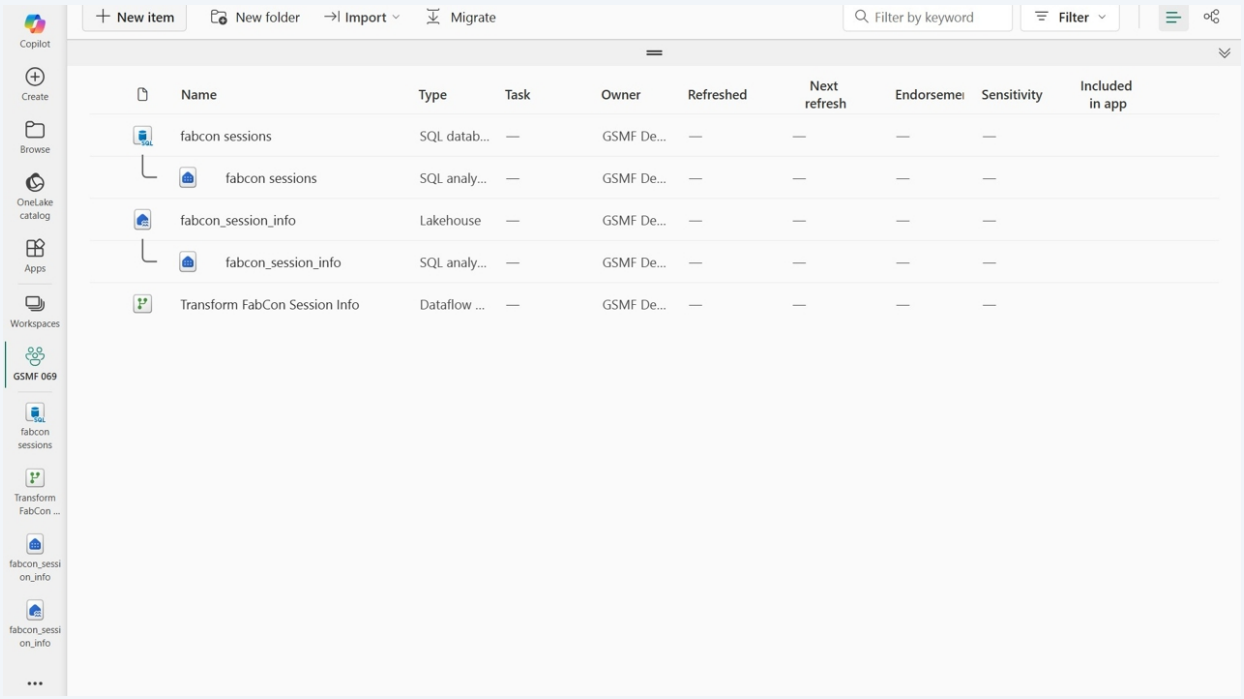
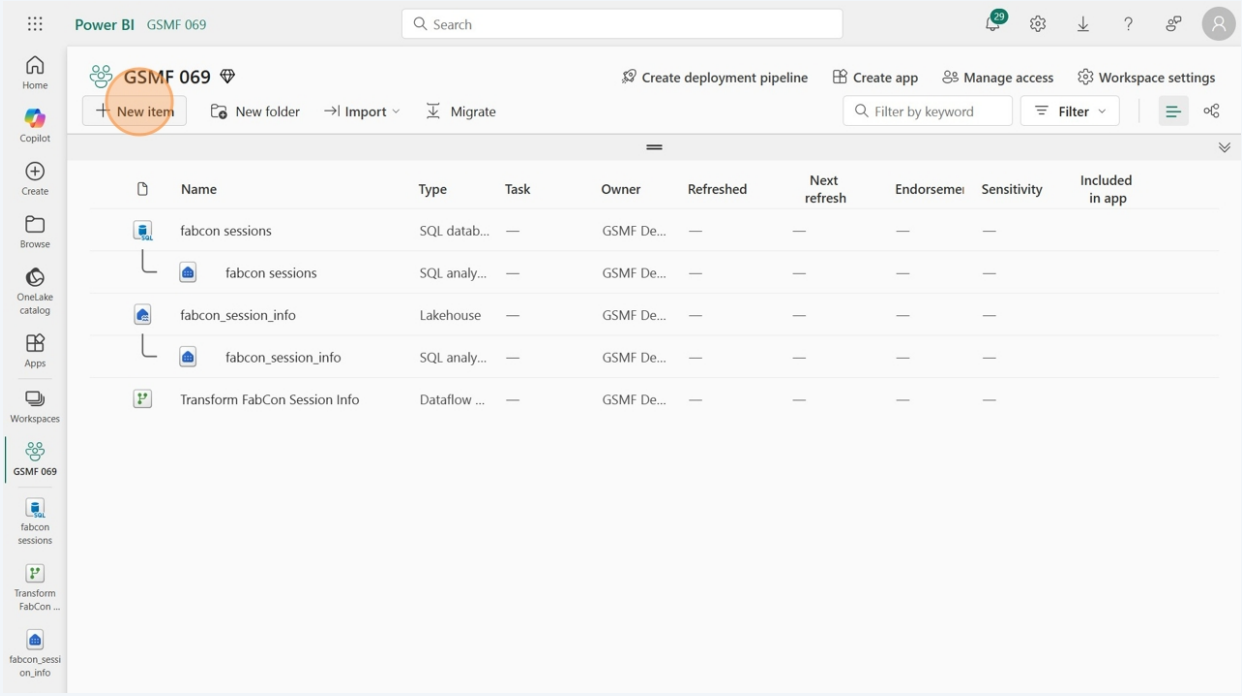
3b. Create and Test a Fabric User Data Functions

Create a new function

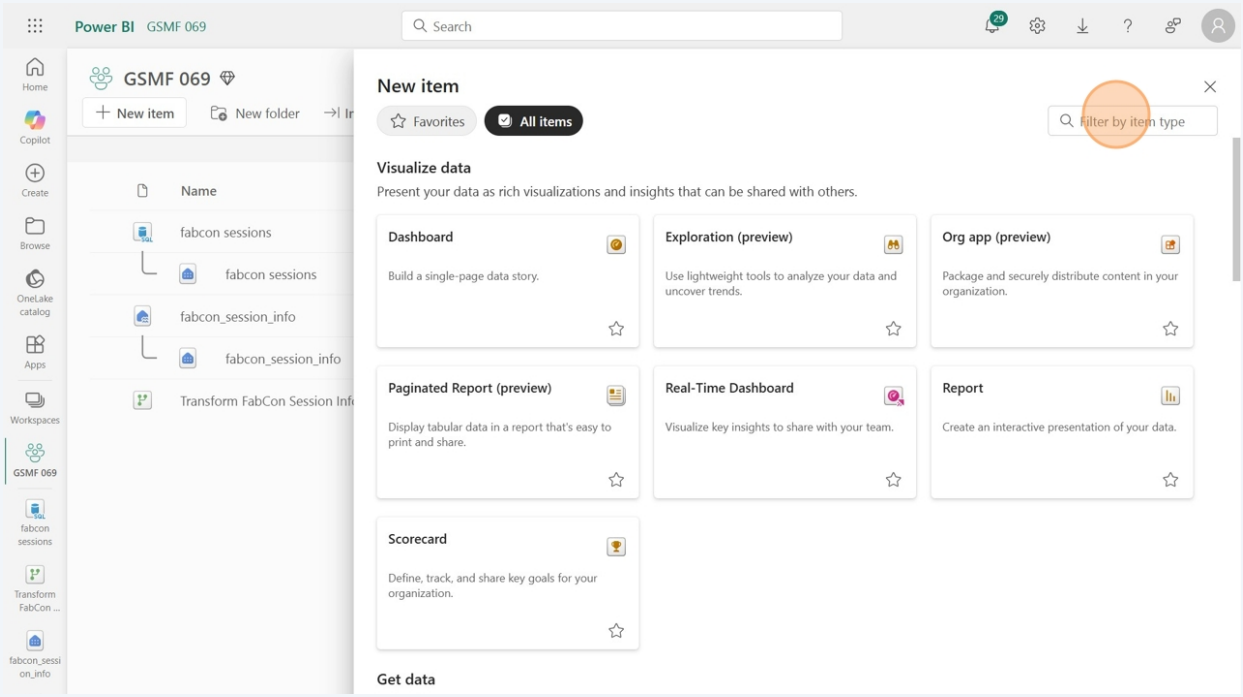
1 Navigate to your workspace. Notice how many items you've built so far! A lakehouse, and Dataflow Gen2 and a SQL database. Now we'll create a User Data Function, which we'll use to write data to our SQL database from a Power BI report.



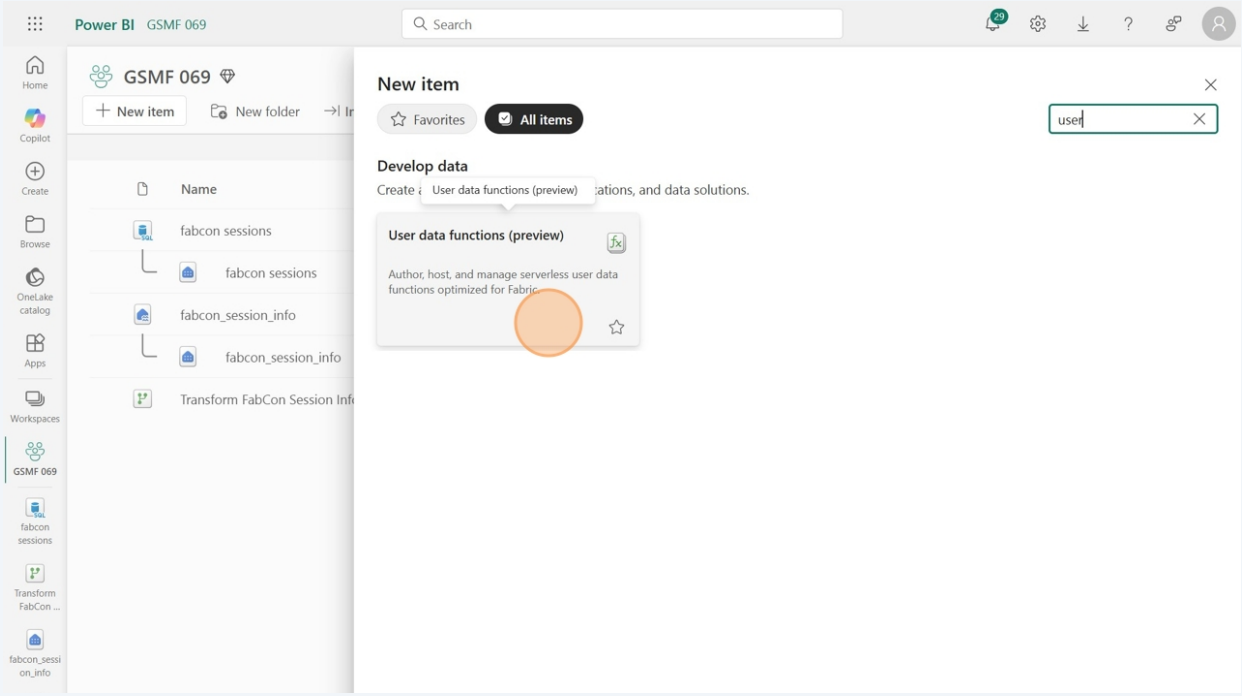
2 Click "New item"



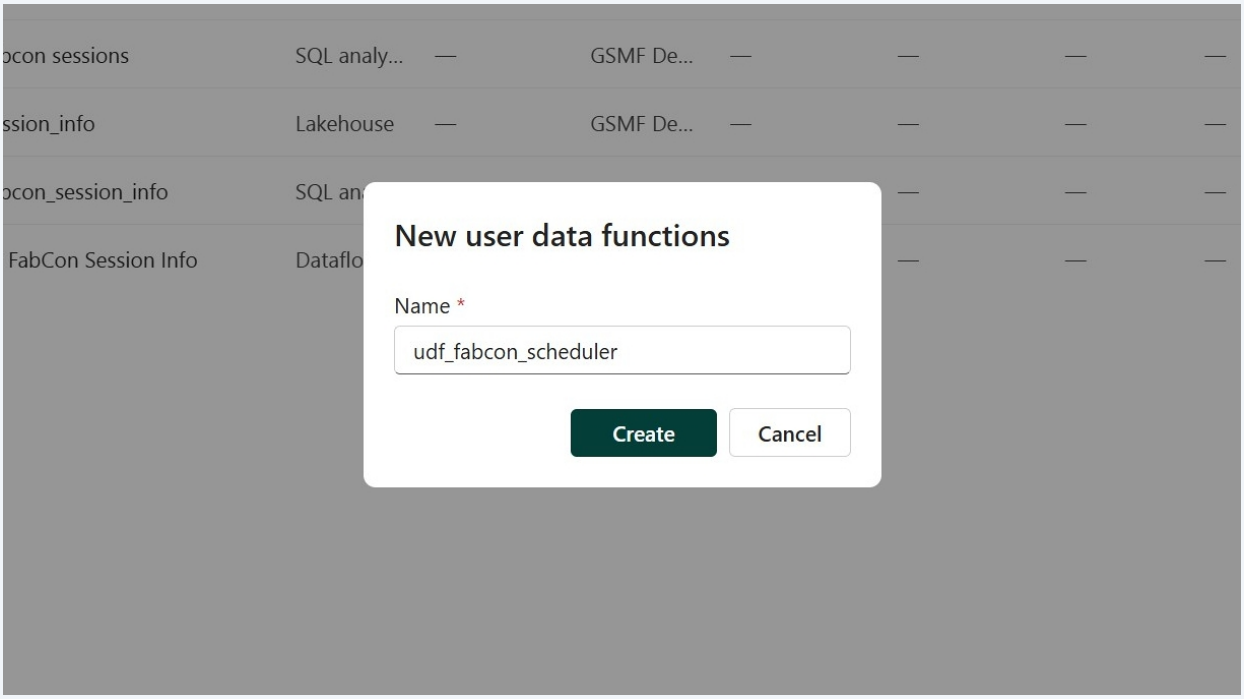
3 Click the "Filter by item type" field and type "user" to easily locate the user data function.



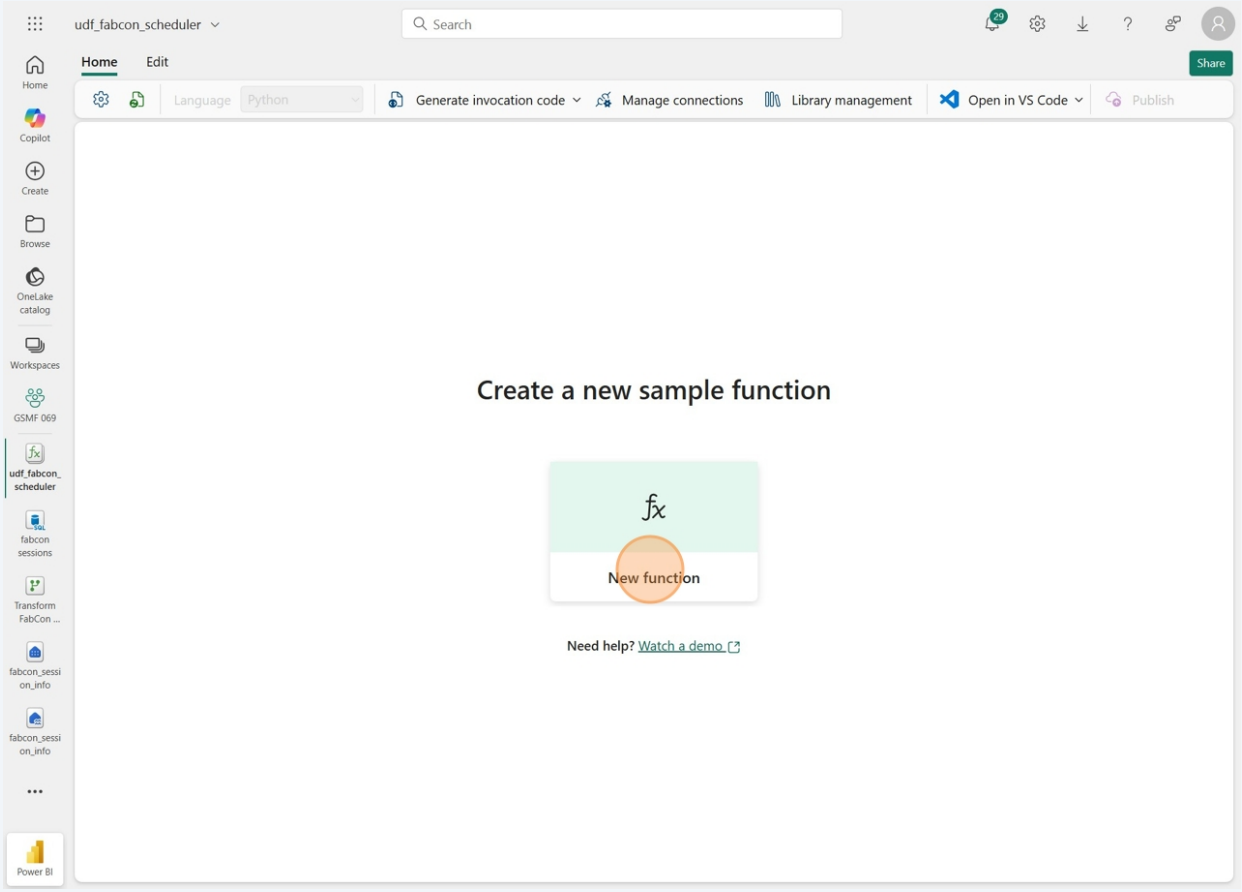
4 Click the "User data functions (preview)" tile.



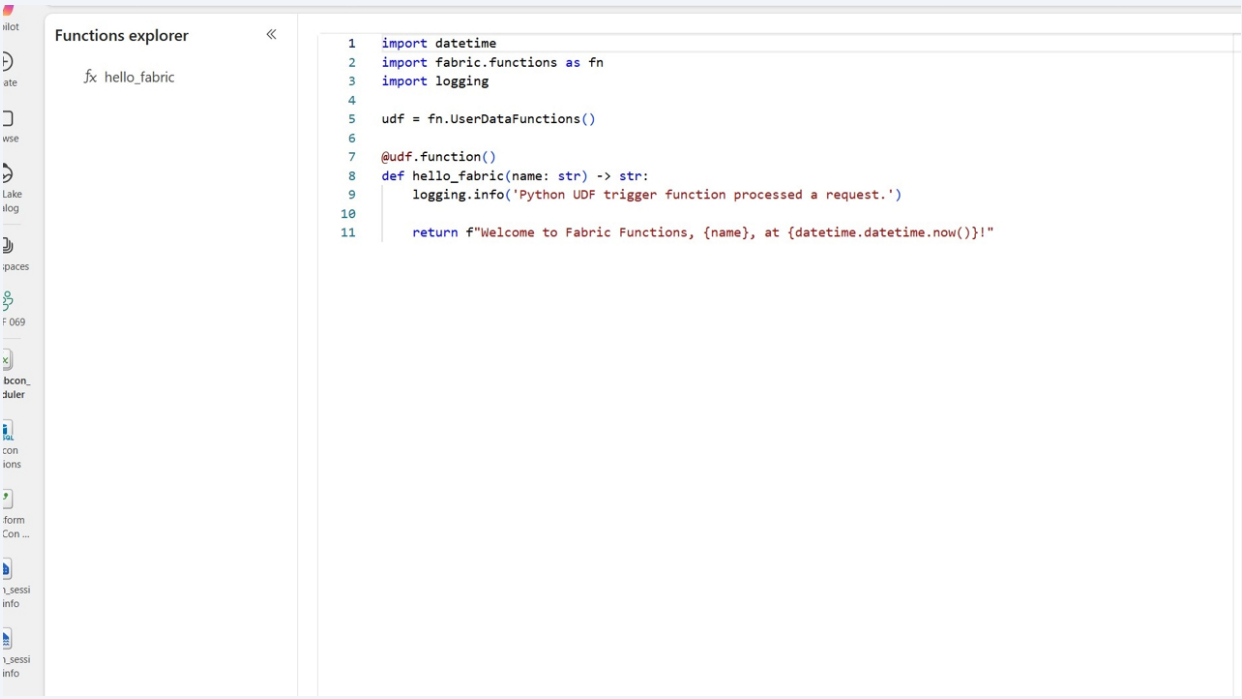
5 Give the function a name like "udf_fabcon_scheduler", for example, and click the "Create" button.



6 Now you're in the user data function editor. Click the big “New function” button in the middle of the canvas.

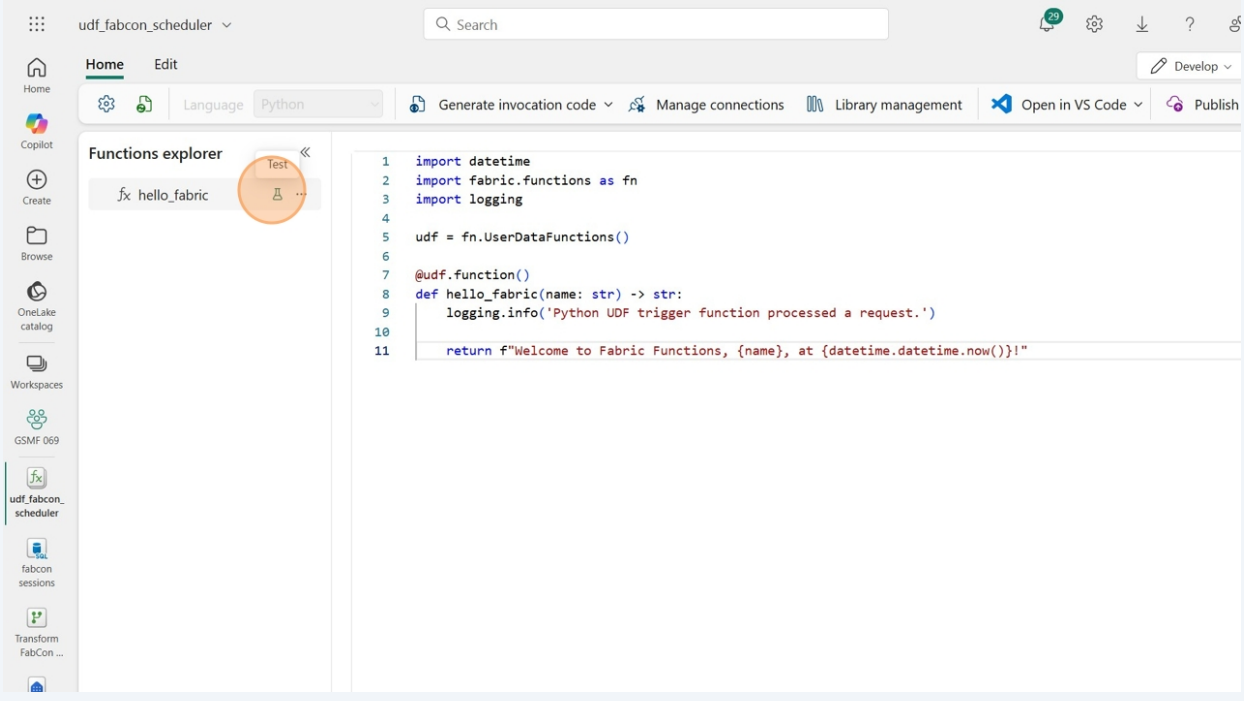


7 A sample function is automatically created. It doesn't do much other than return a message based on an input, but it gives you an idea of what the code should look like. We'll test this sample function to see what happens.

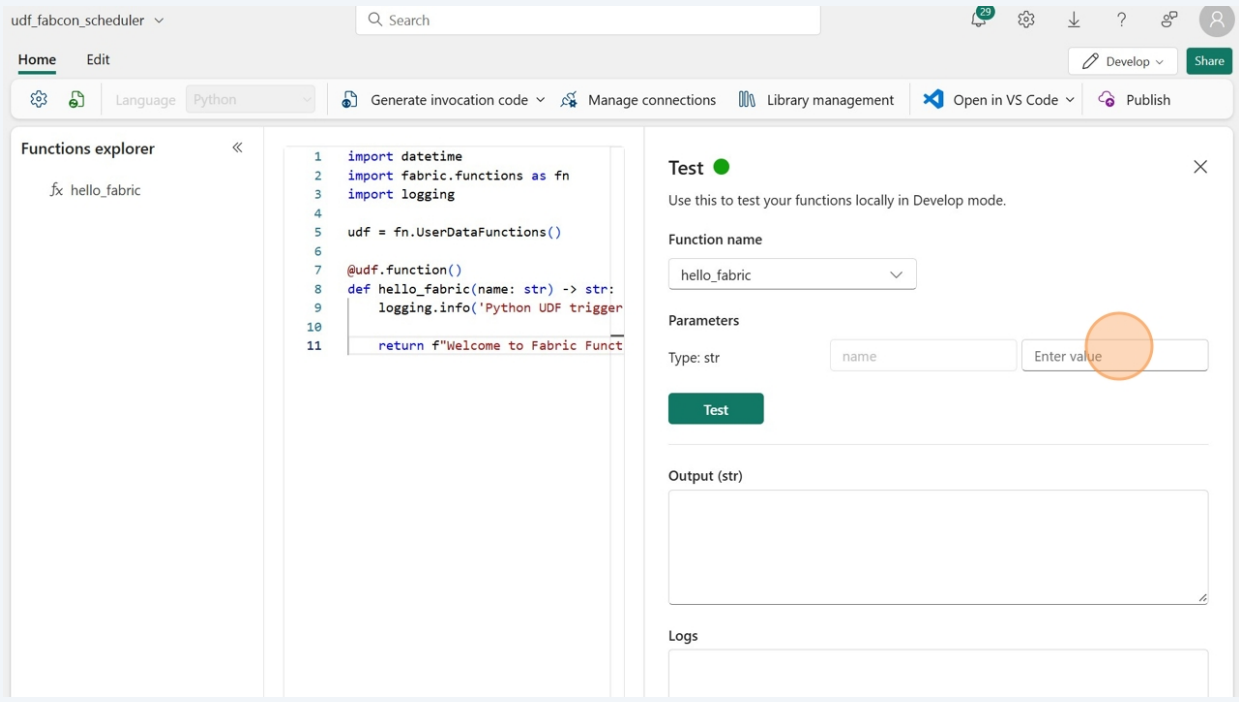


Test the sample function

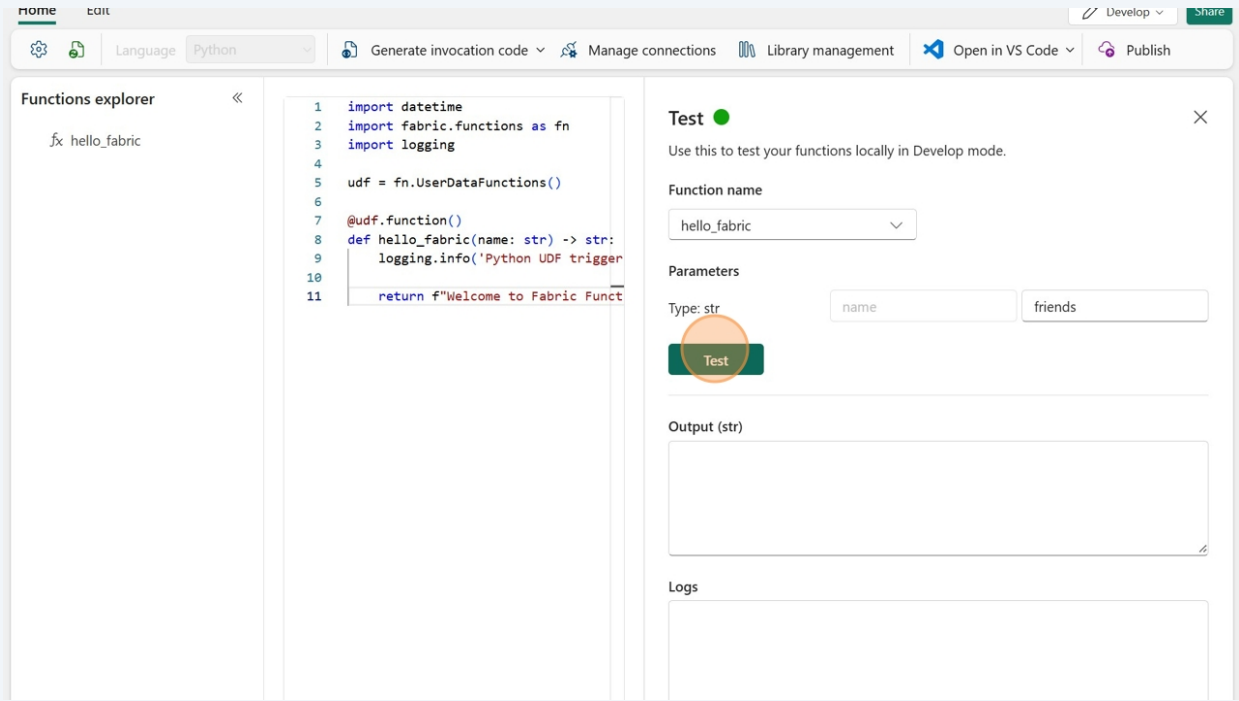
8 Hover over the “hello_fabric” function and click the little test beaker to test the function.



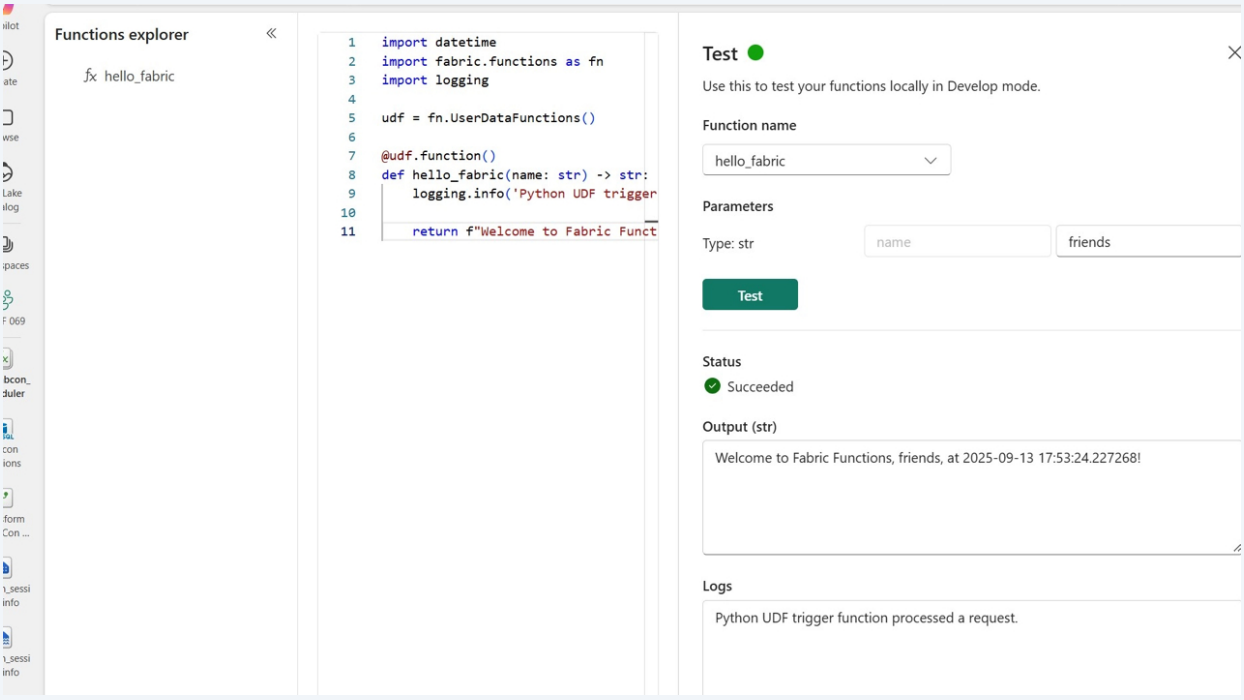
9 Click the "Enter value" field and enter you name.



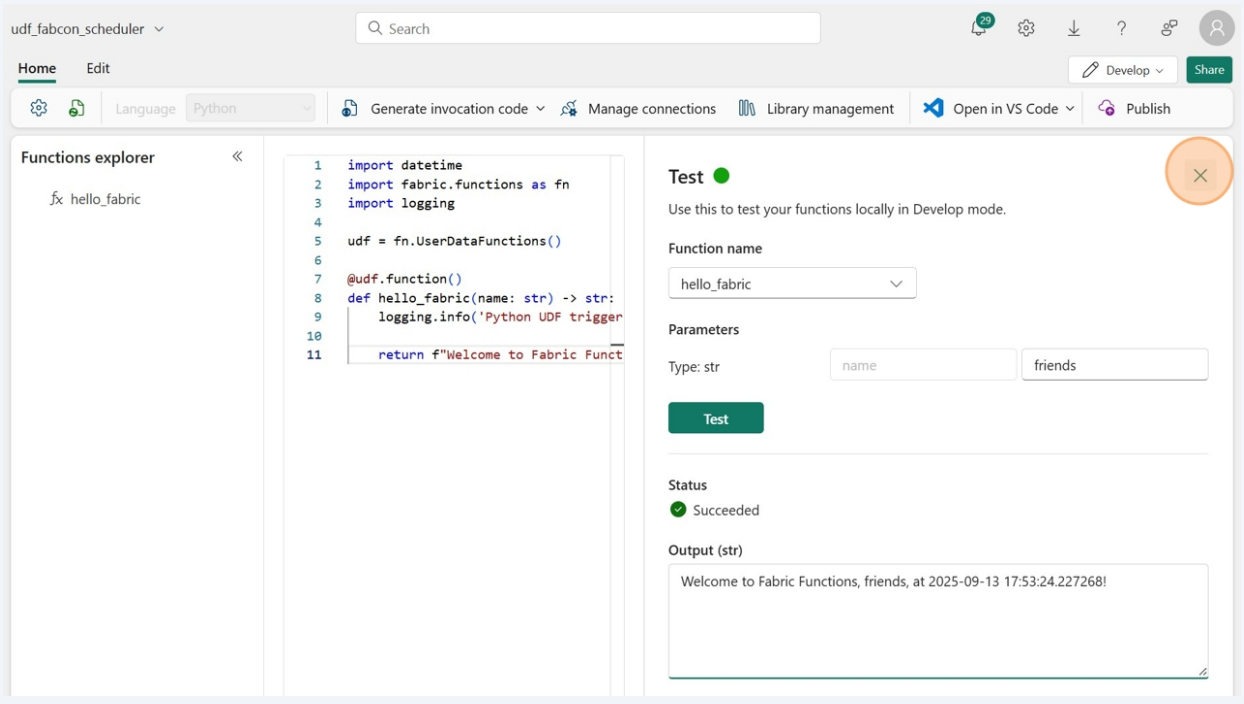
10 Click the "Test" button. (make sure it gets clicked and doesn't just move your cursor)



11 After a few seconds you should see the output and any logging information. It may take a few seconds for it to kick off, so be patient.

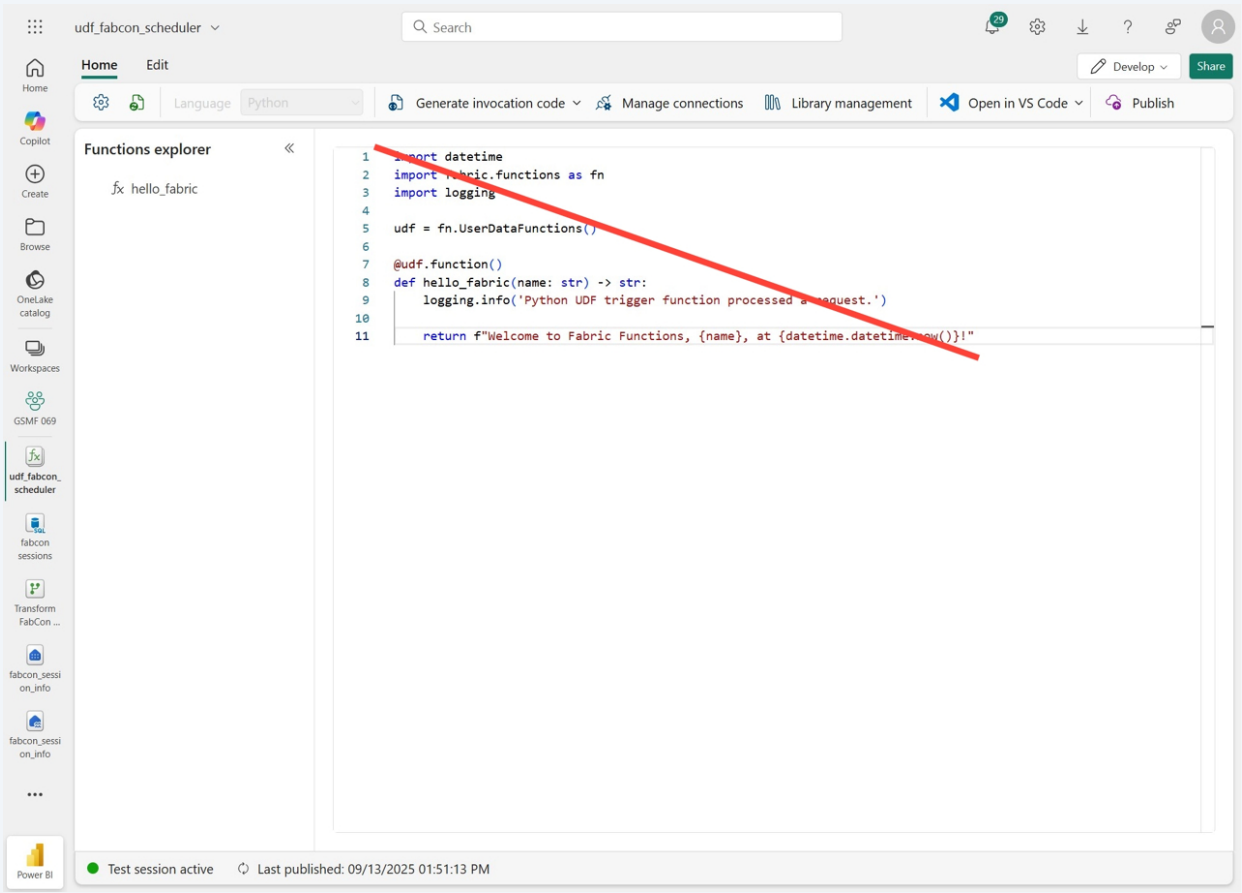


12 Close the test pane by clicking the "X" in the top right.



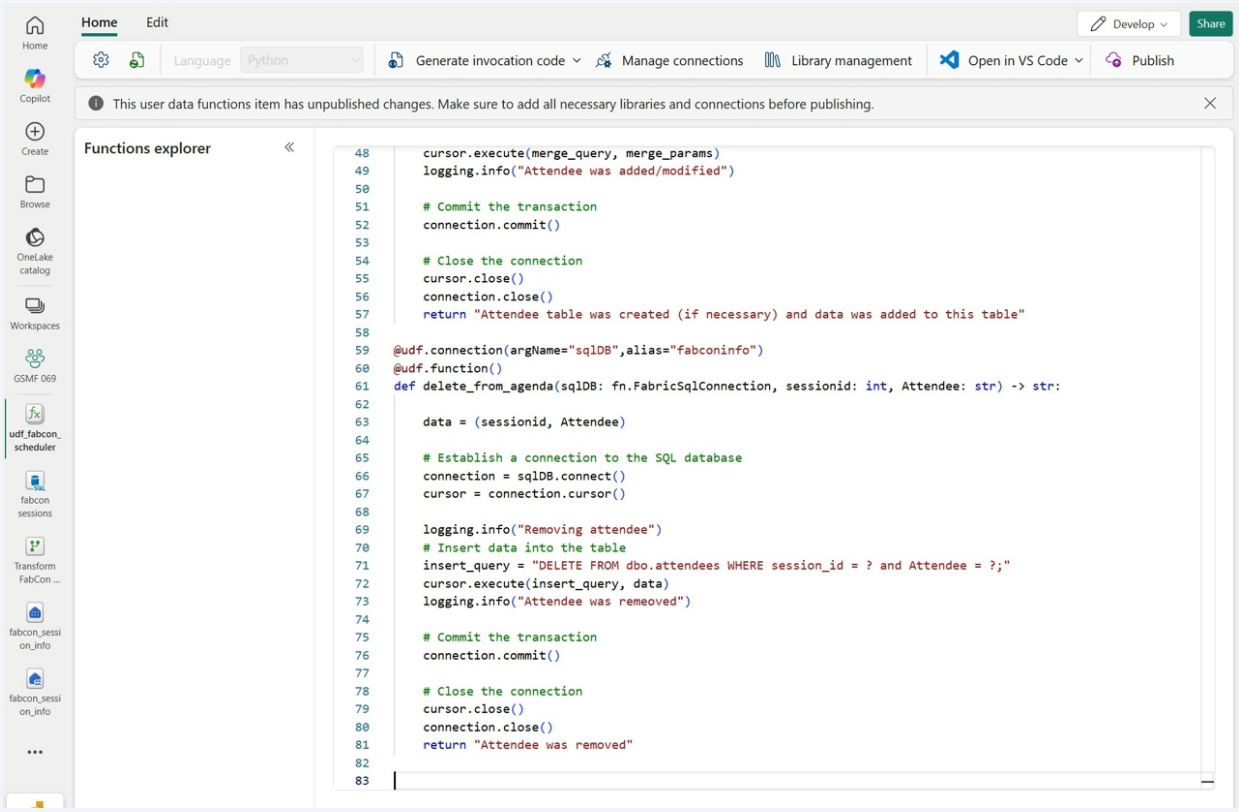
Enter the code for the functions

13 Now we'll replace the sample function with a real function. Highlight all the code the delete it so there is no text left.

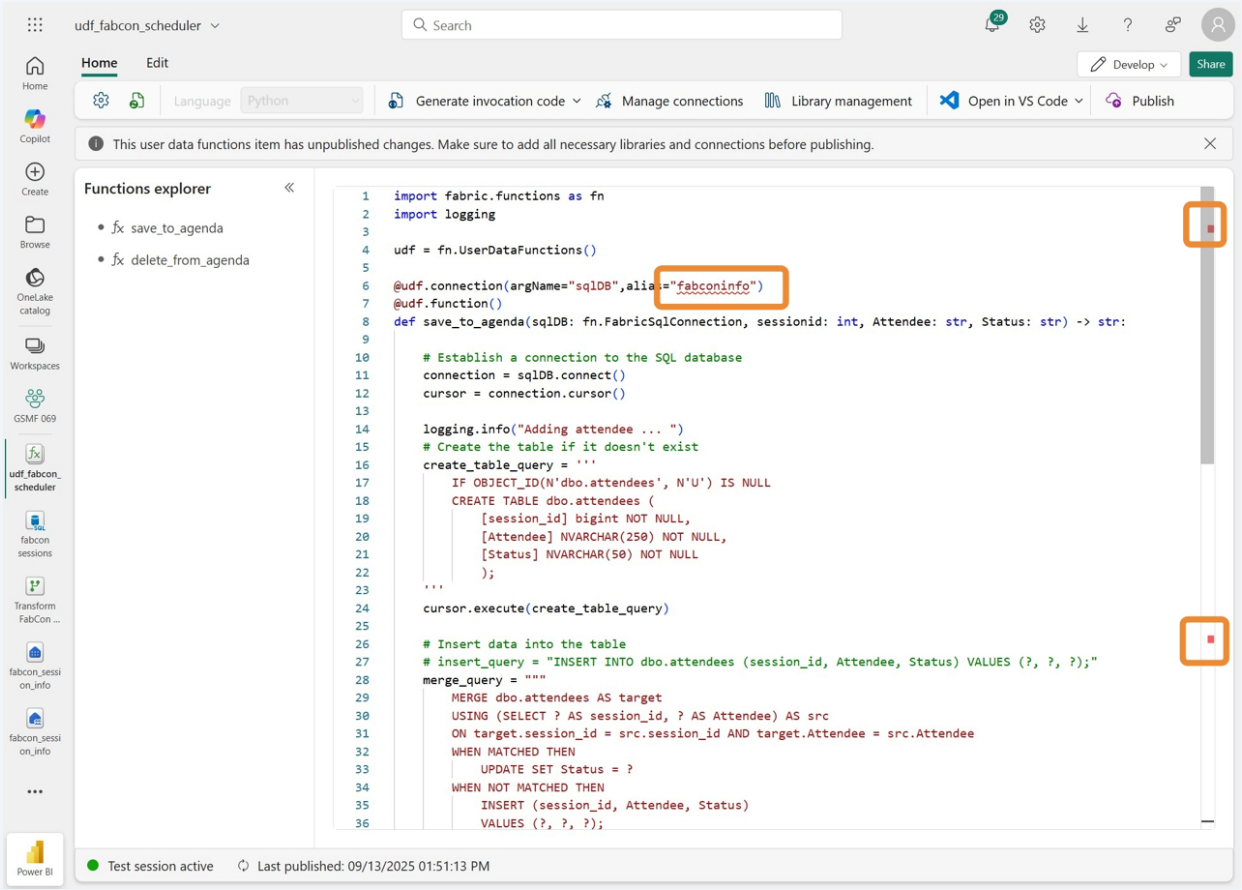


14 Open the UDF.txt file from the resources section and copy all the code. Paste it into the UDF editor.

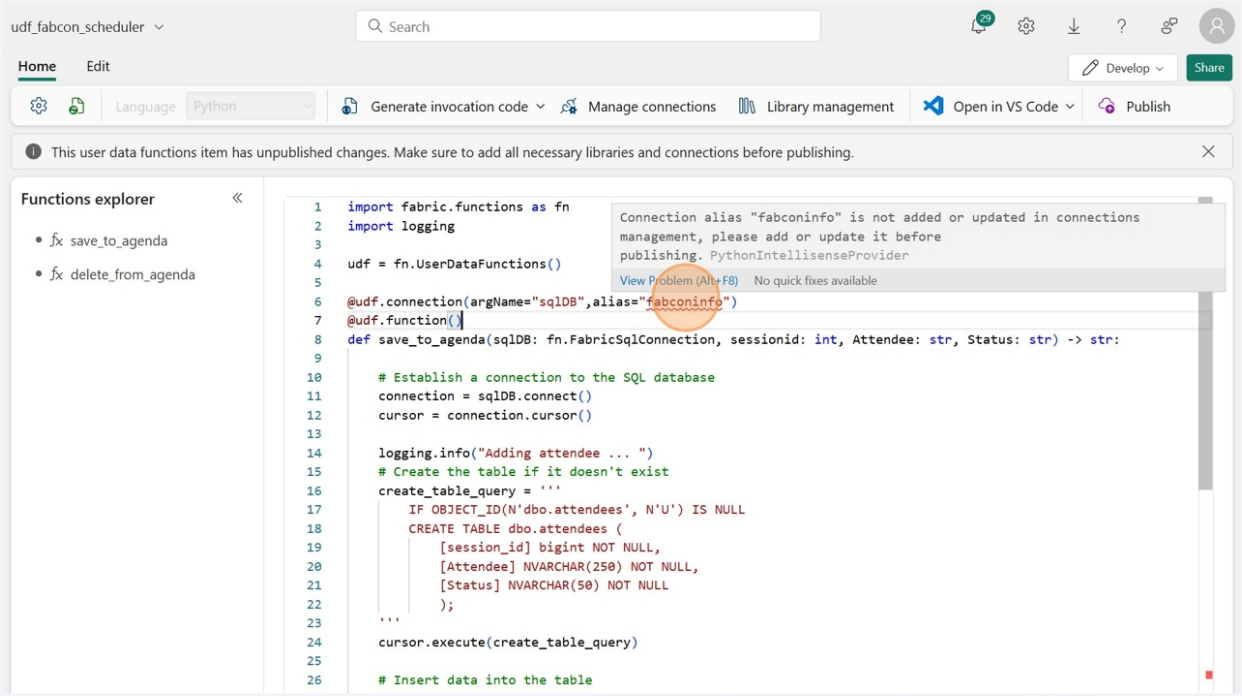
15 Your function should now look like this. In fact, we have two functions (which you can see in the Functions explorer on the left). One for adding a session to your agenda, and one for deleting it from your agenda. Take a look at the code to get an idea of how it works.



16 Notice there are two places where the text has a squiggly red line. You can see in the scrollbar on the right side that there are two little red squares indicating errors.

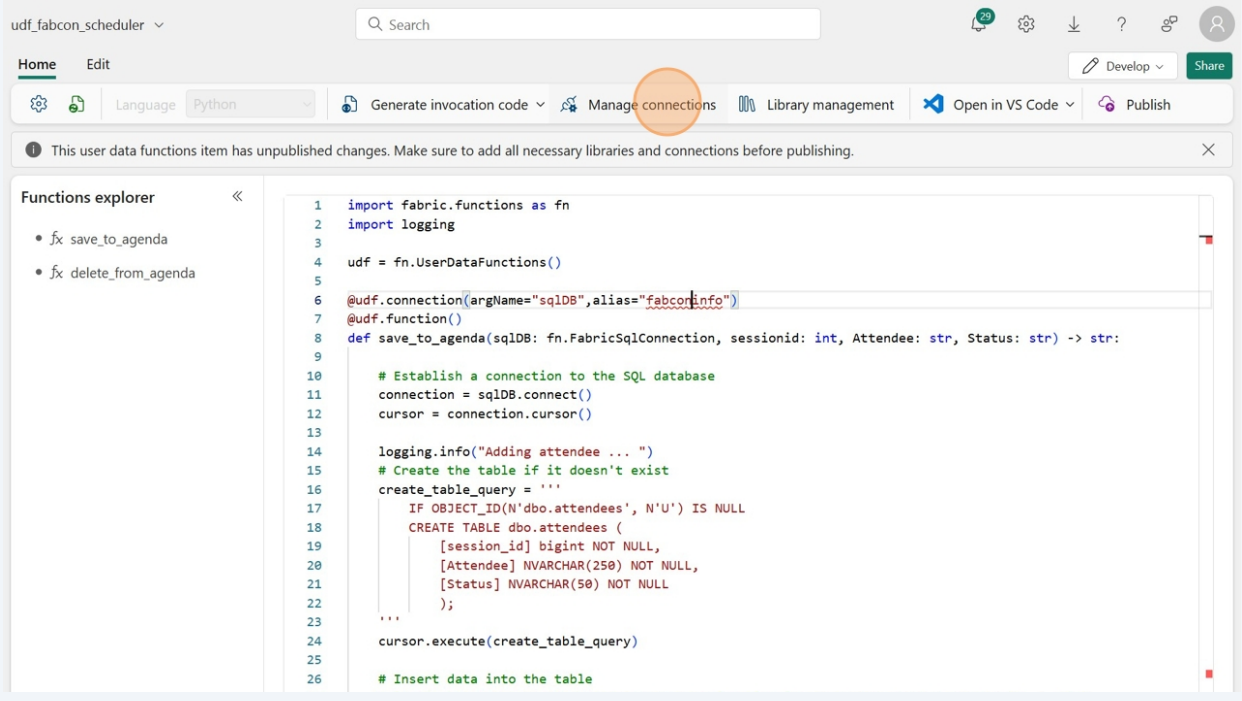


17 If you click the error, you'll see a message telling you that we are referring to a connection that dosen't exist. We need to create a connection to our SQL database for this function to use as the connection.

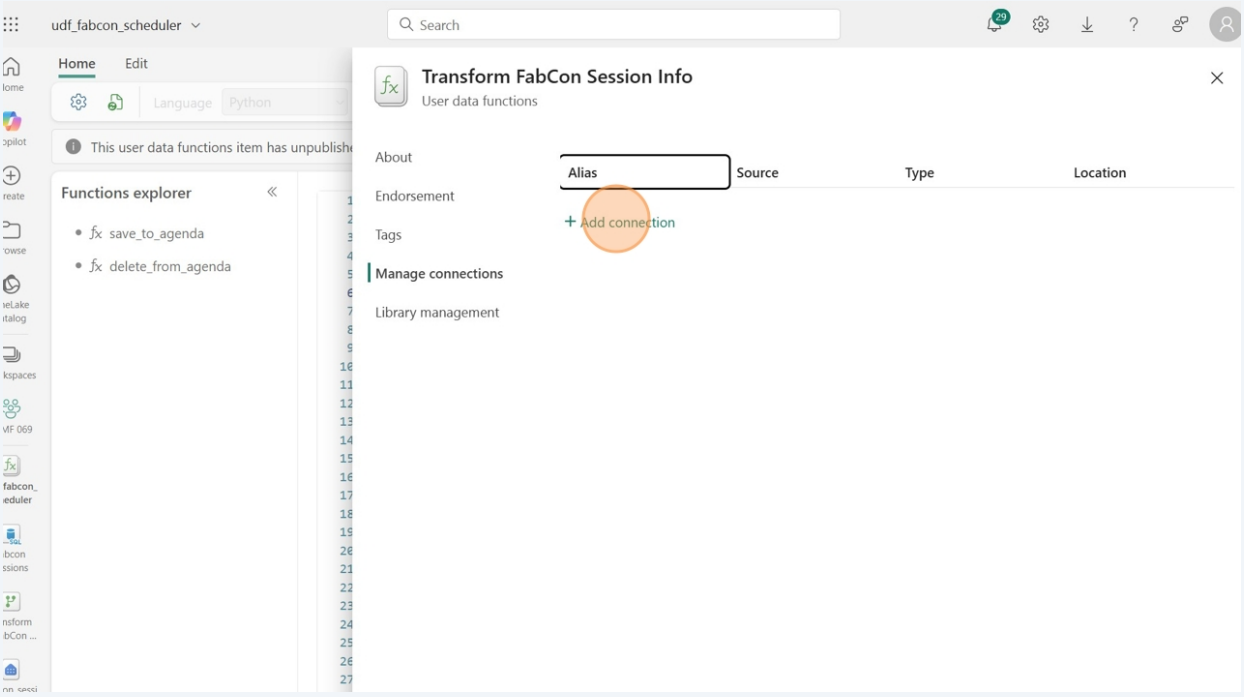


Create a connection to the SQL database

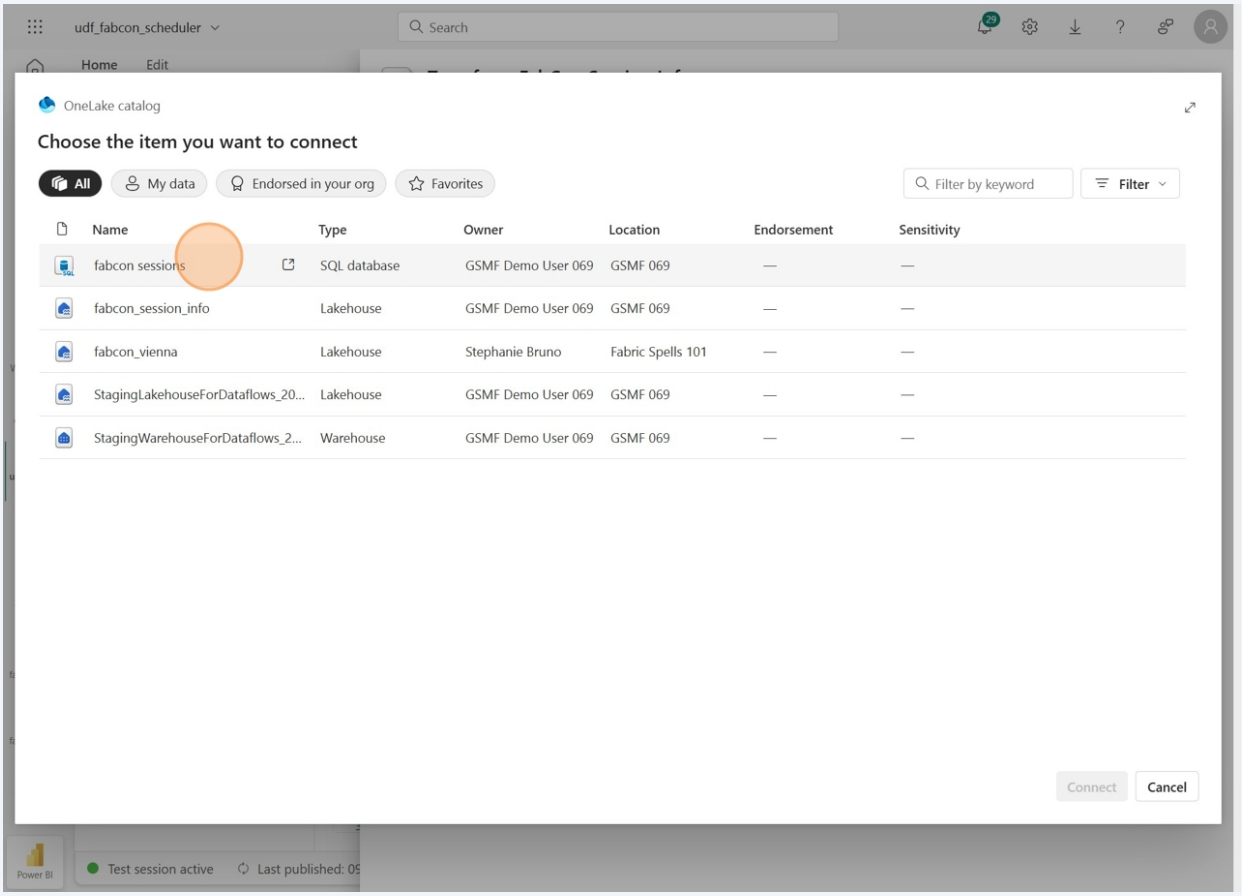
18 Click "Manage connections"



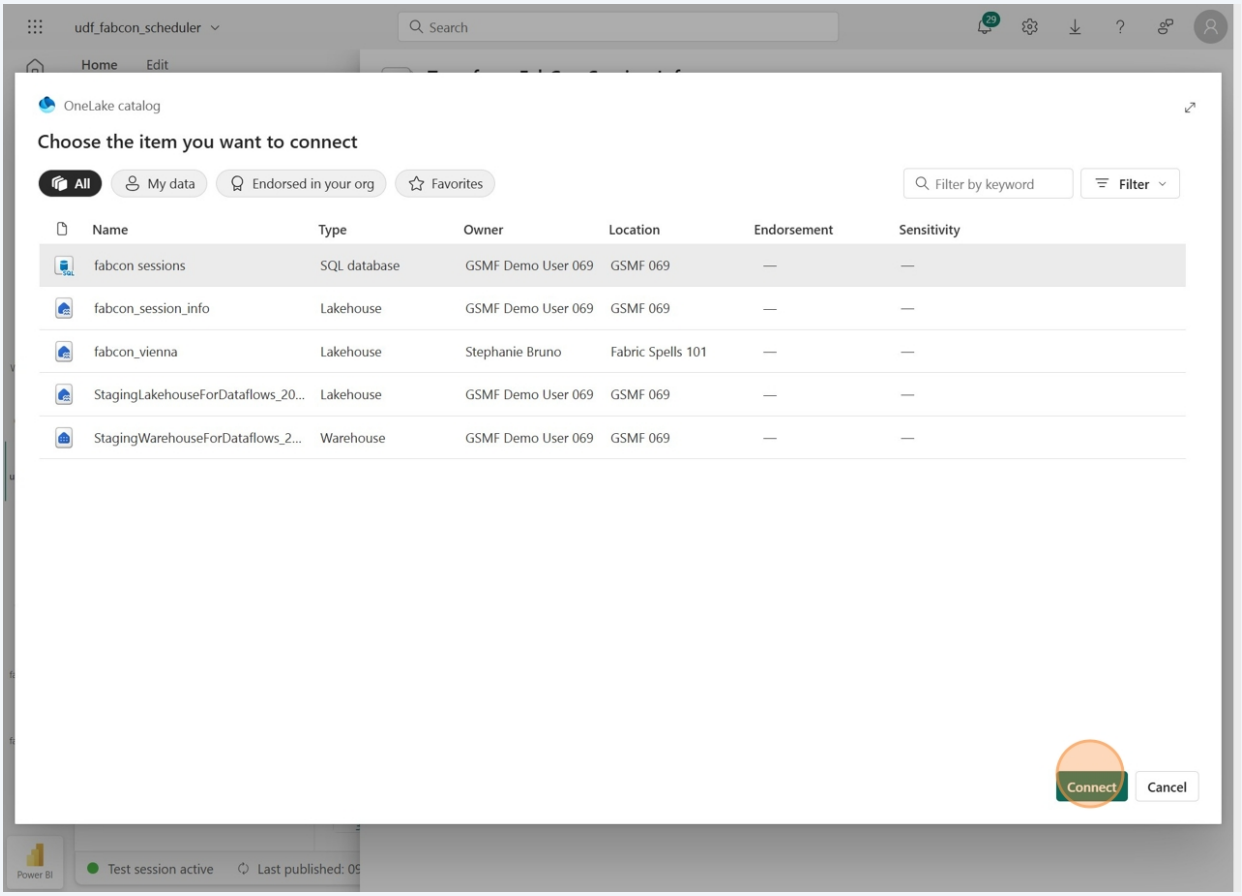
19 Click "Add connection"



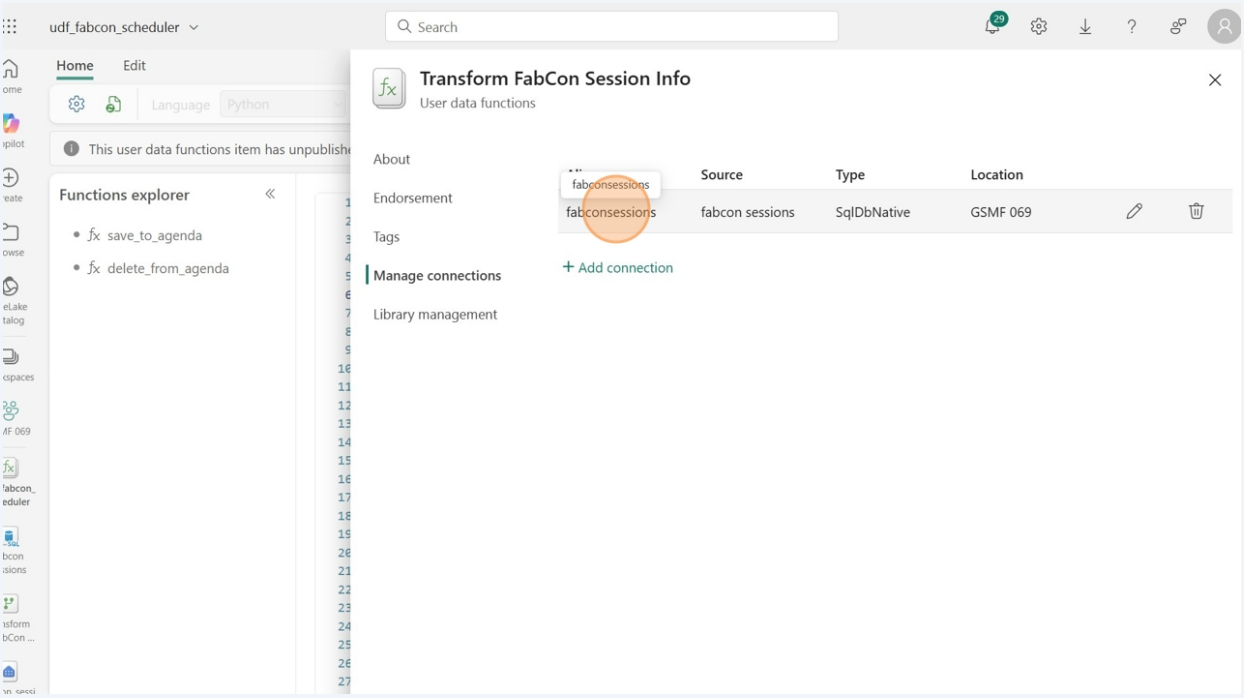
20 Click on the SQL database you created in the previous lab. This is the one we want to connect to.



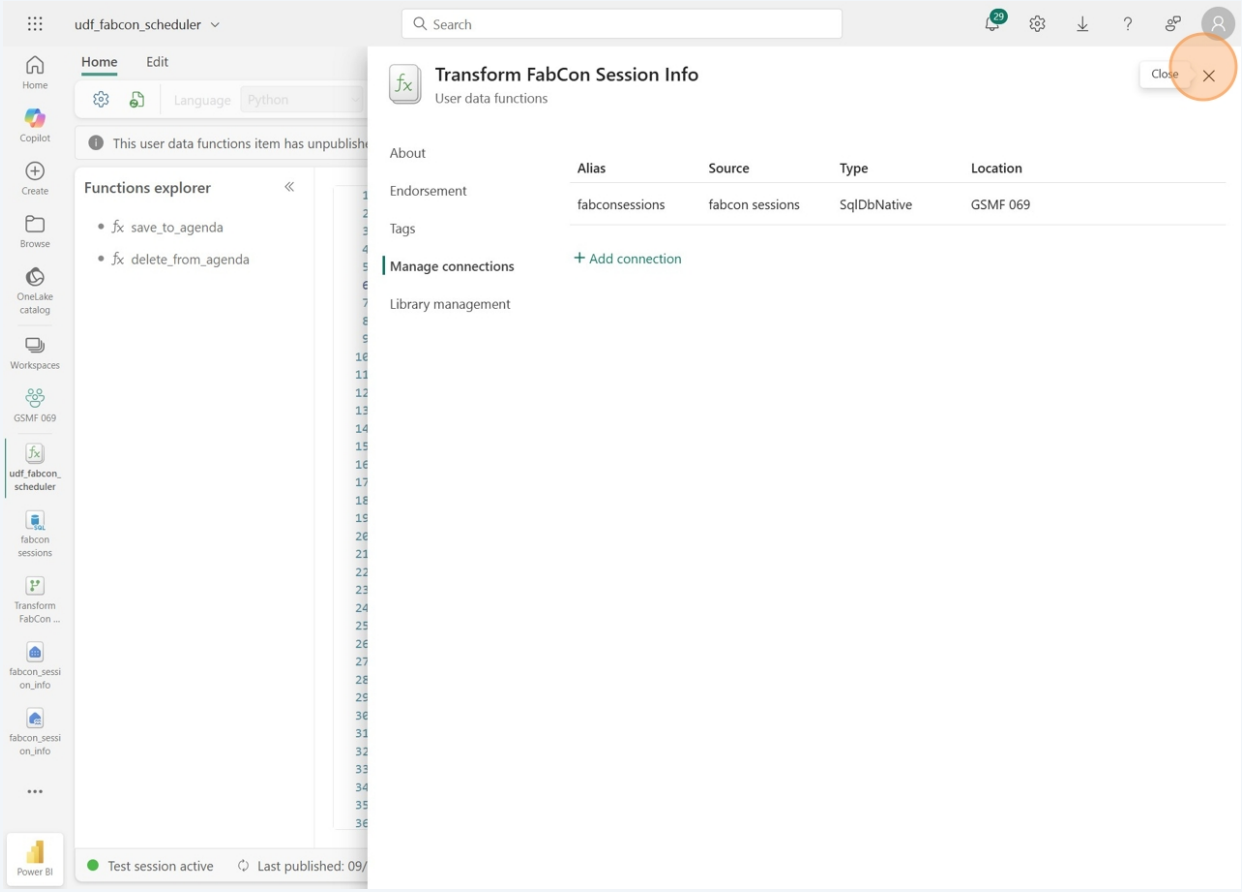
21 Click "Connect"



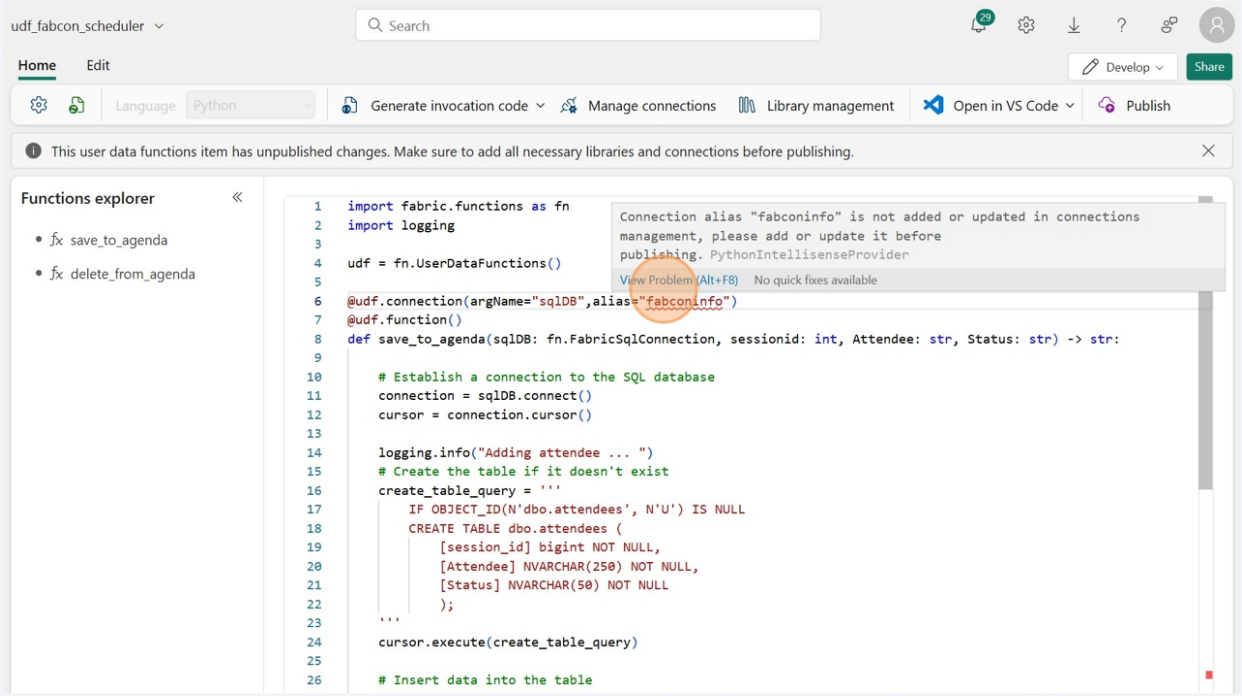
22 You should now see one connection listed. The source is your SQL database and the type is SqlDbNative. The location is your workspace. It automatically created an alias for the connection based on your SQL database's name. You can either edit that alias or accept it as-is. Either way, the alias is what we need the code to refer to in the two lines using the connection. Copy the alias name so you can update the code with this value.



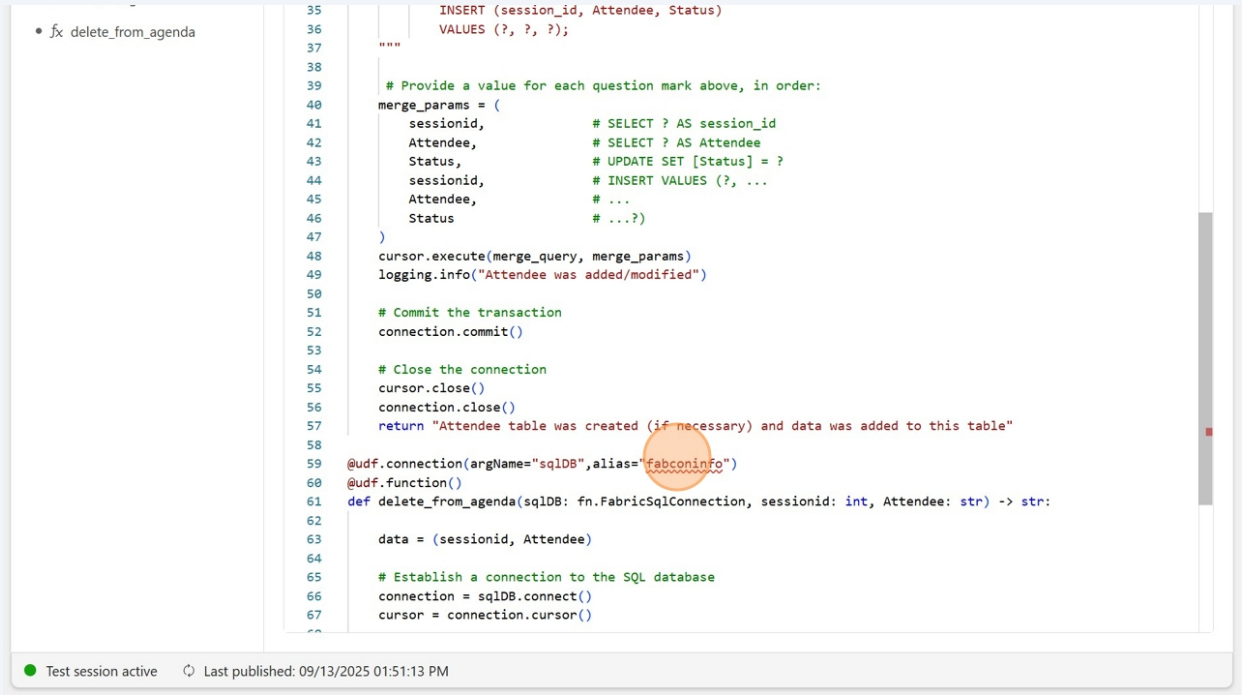
23 Click the "X" to close the connections pane.



24 Double-click "fabconinfo" on line 6 of the code and replace it with whatever your connction's alias name is.

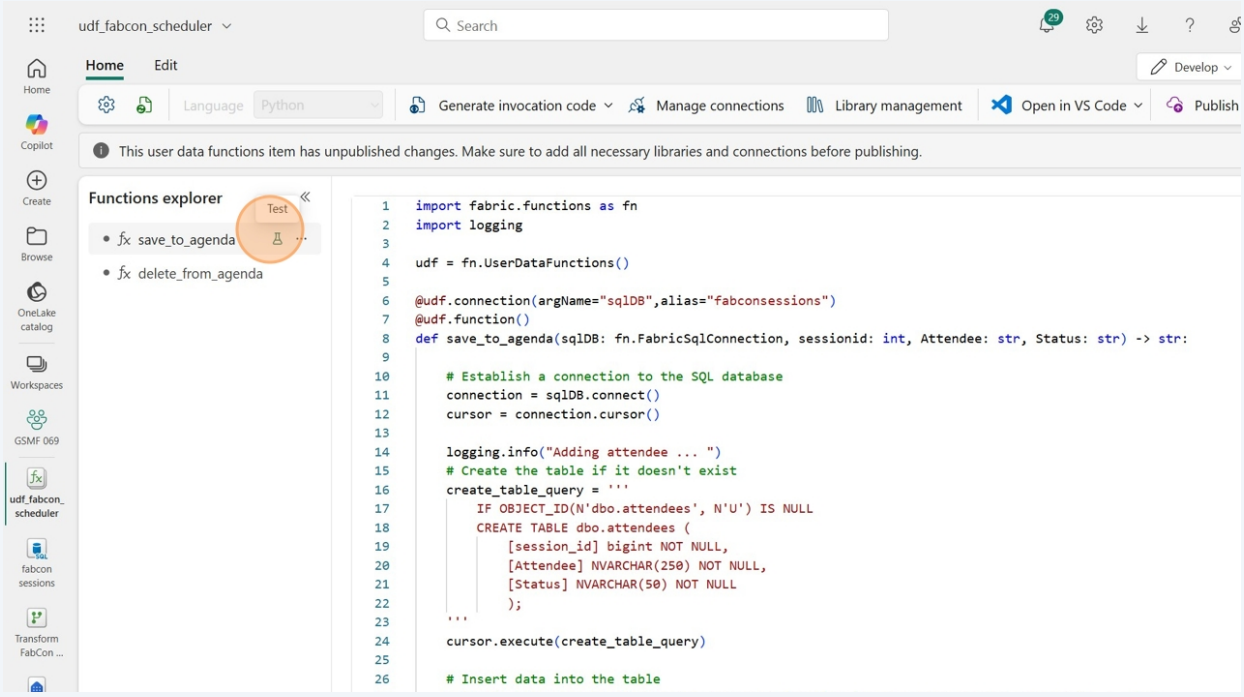


25 Do the same on line 59. After these two replacements, you shouldn't see the little red marks on the right-side scroll bar anymore.



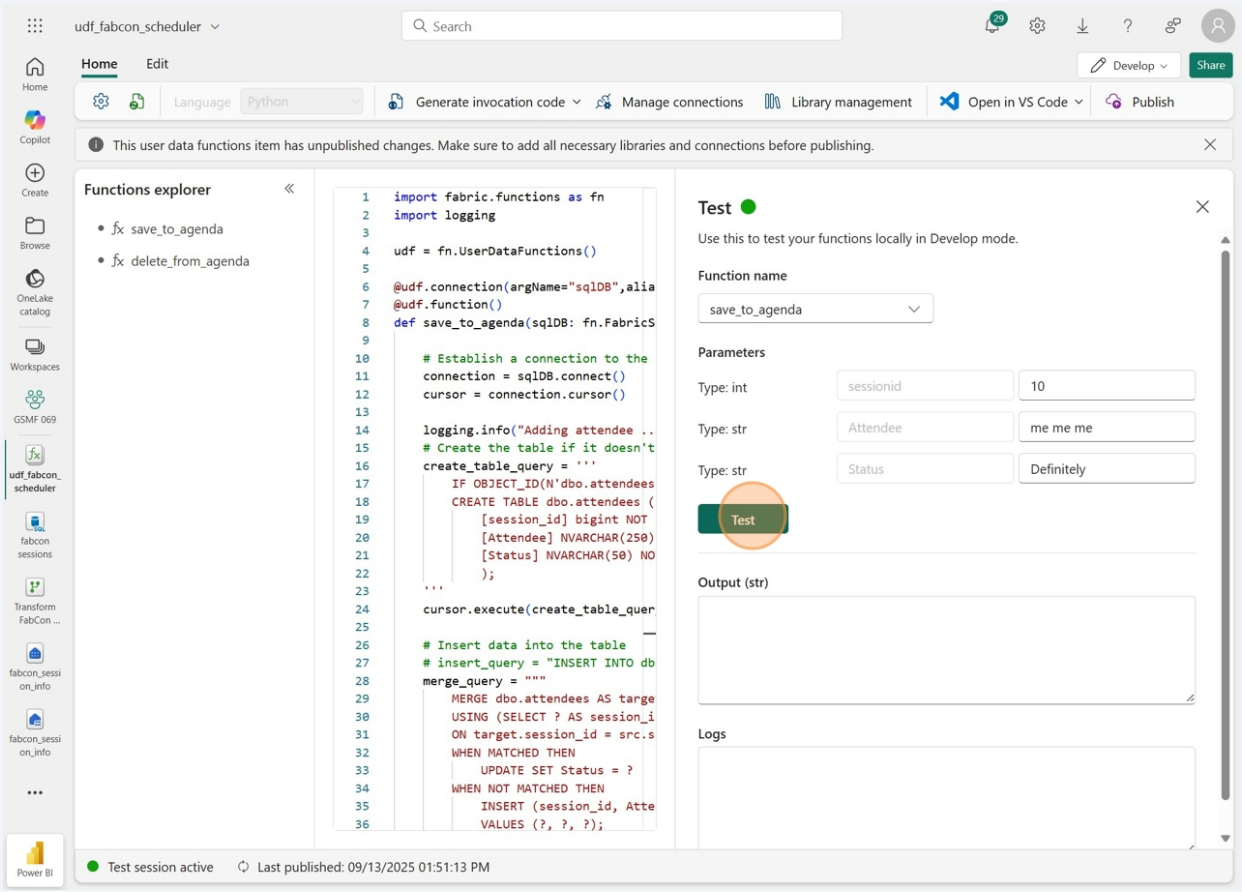
Test a function

26 Now we're ready to test! Click the "Test" icon next to the "save_to_agenda" function name.

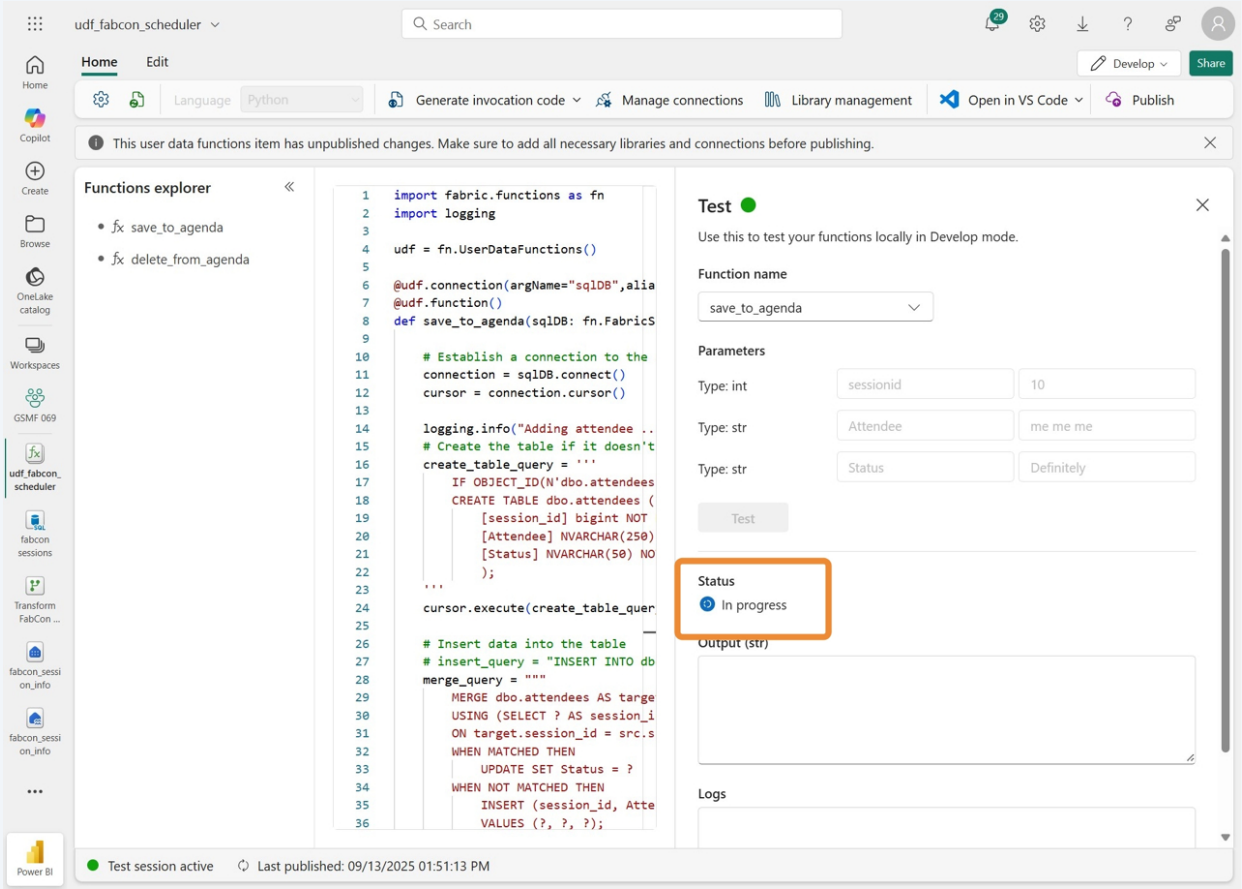


27 The "save_to_agenda" takes three parameters: sessionid, Attendee, and Status. For testing purposes, it doesn't really matter what you enter, but sessionid must be an integer.

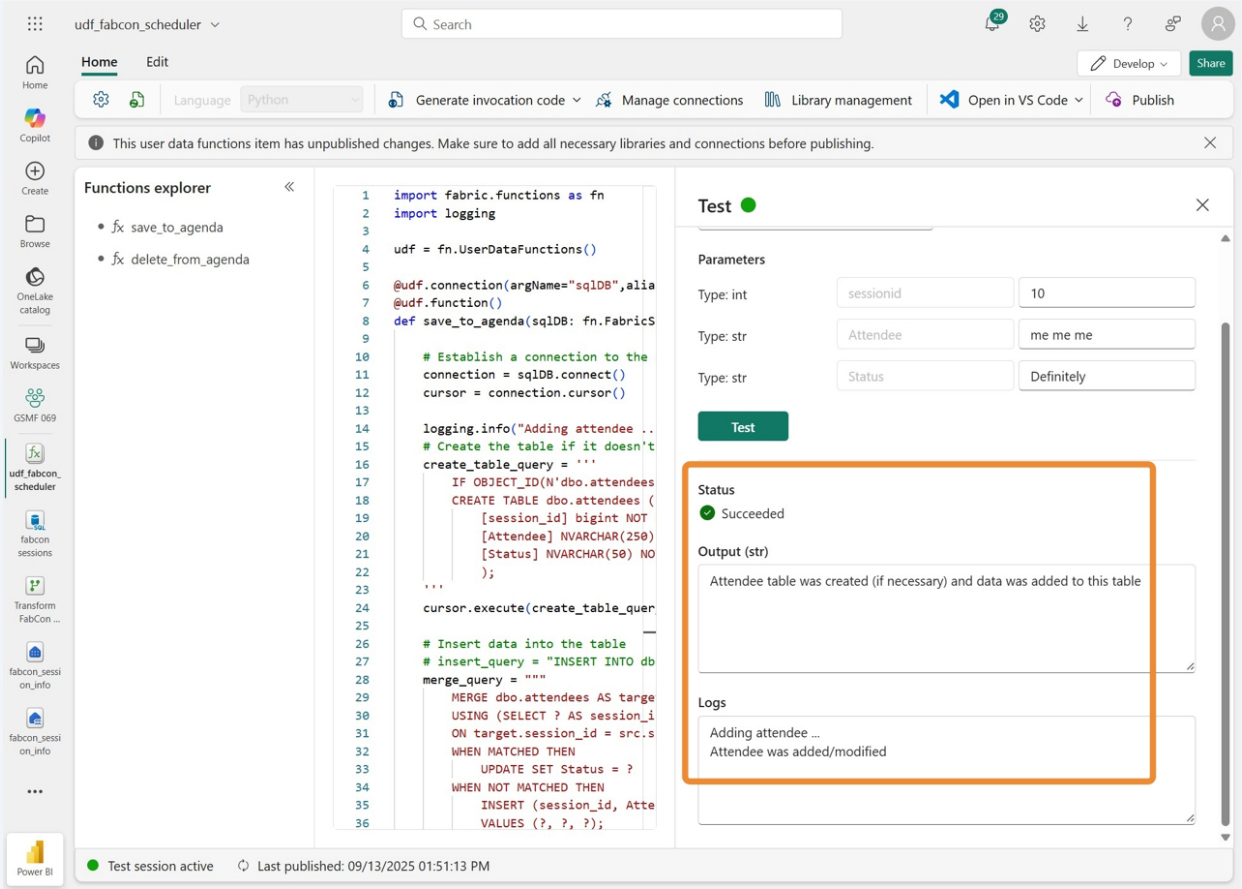
Enter values for each of the three parameters and click the "Test" button.



28 While the function is running you can see the status is "In progress."

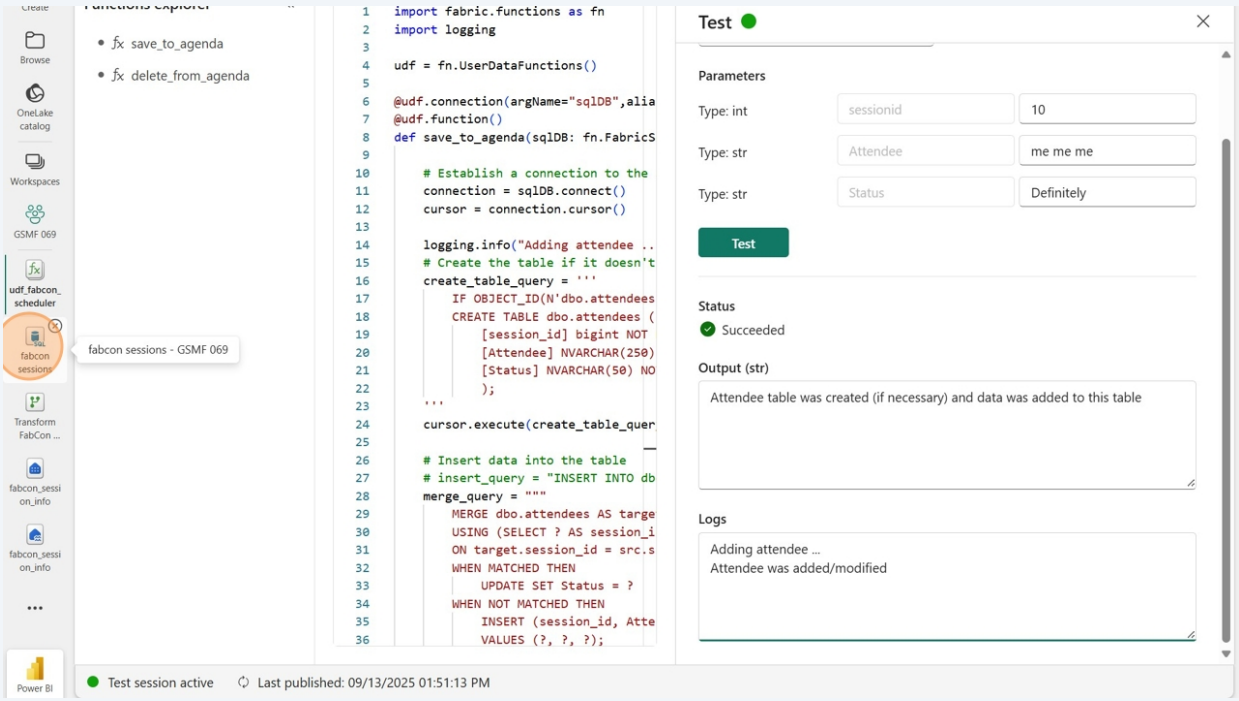


29 After it's finished you can see the output and logs again. We can see that we added some data to the attendee table.

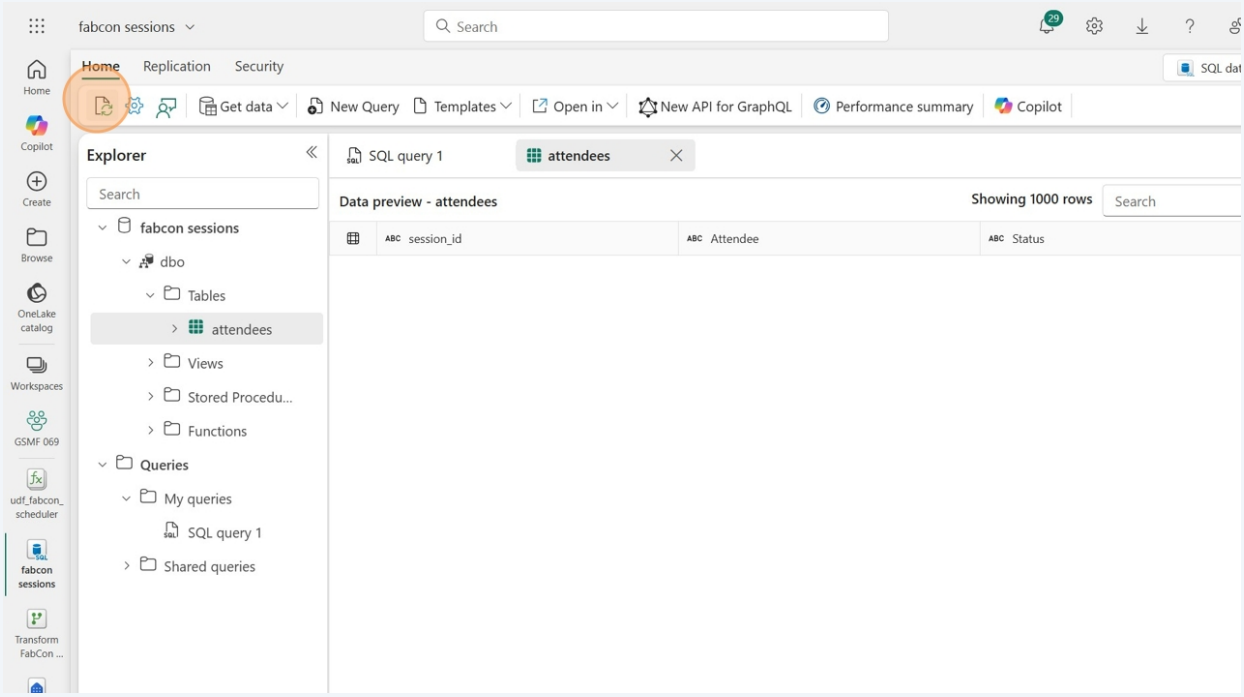


Verify the data in the SQL database

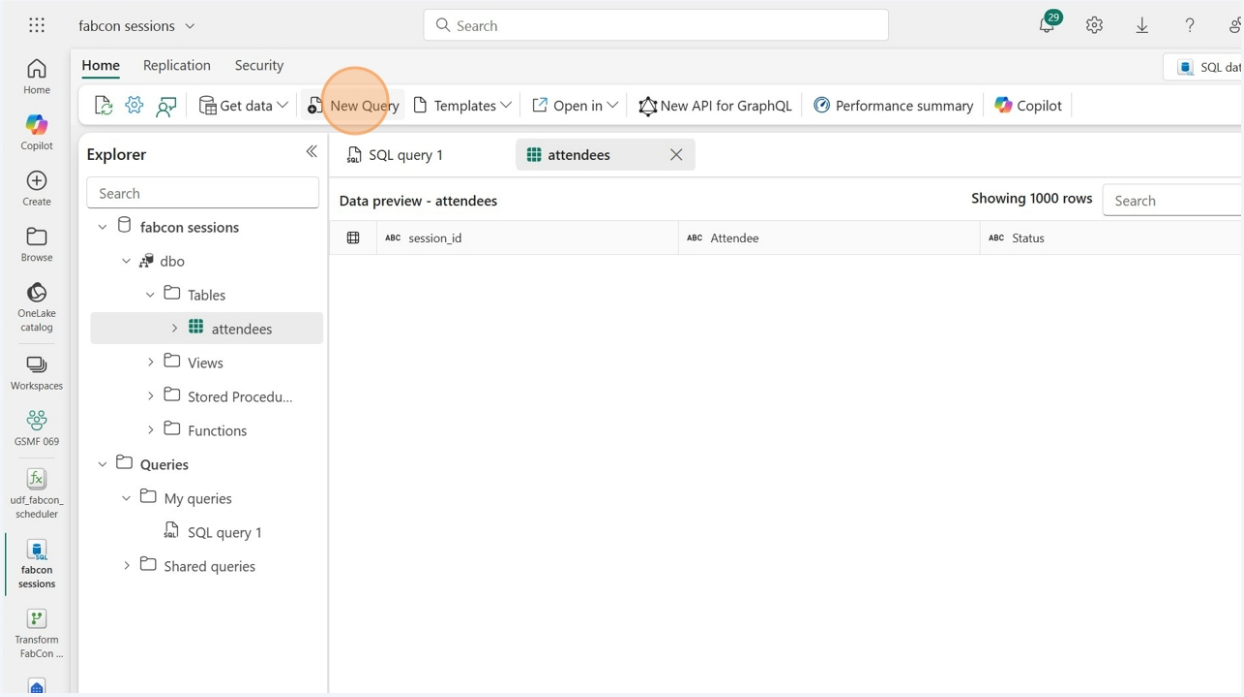
30 Let's go see the data in the SQL database. Go back to the SQL database by clicking on its icon on the left, or by going back to the workspace and clicking on it there.



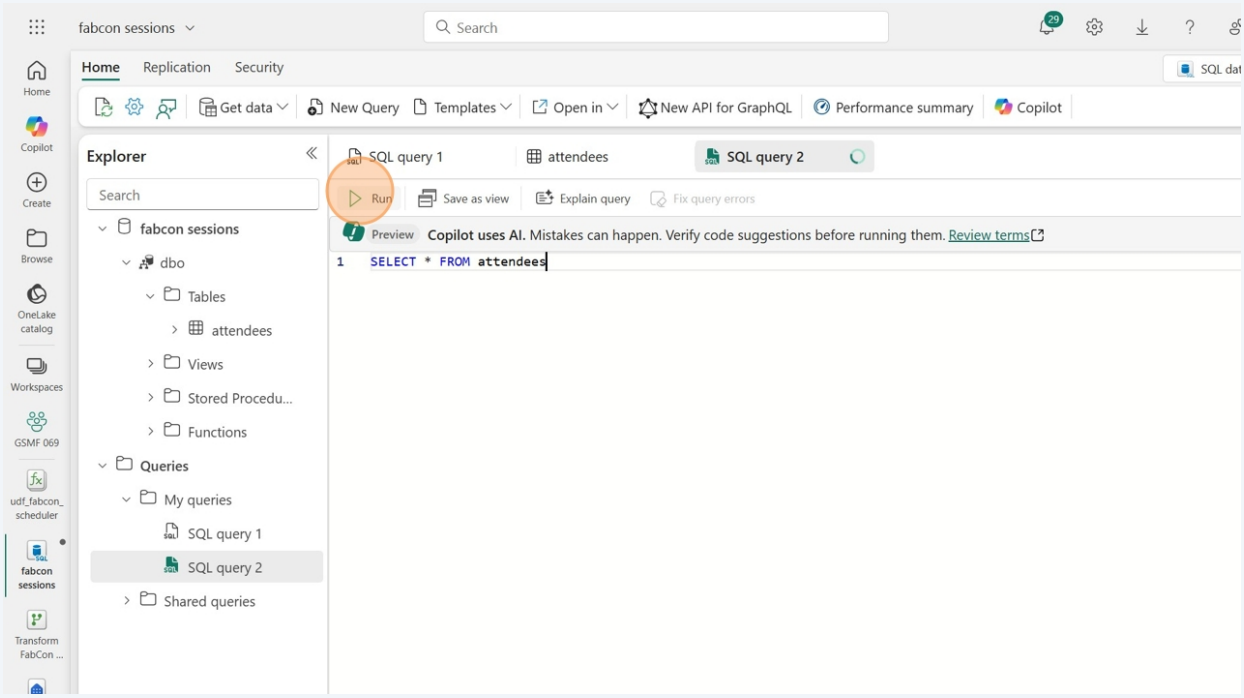
31 Click on the attendees table to see a preview of the data. If you don't see anything yet, click on the refresh button in the Home ribbon or next to the Tables folder. It's possible you might still not see it in the preview (sometimes it takes a few minutes to show changes).



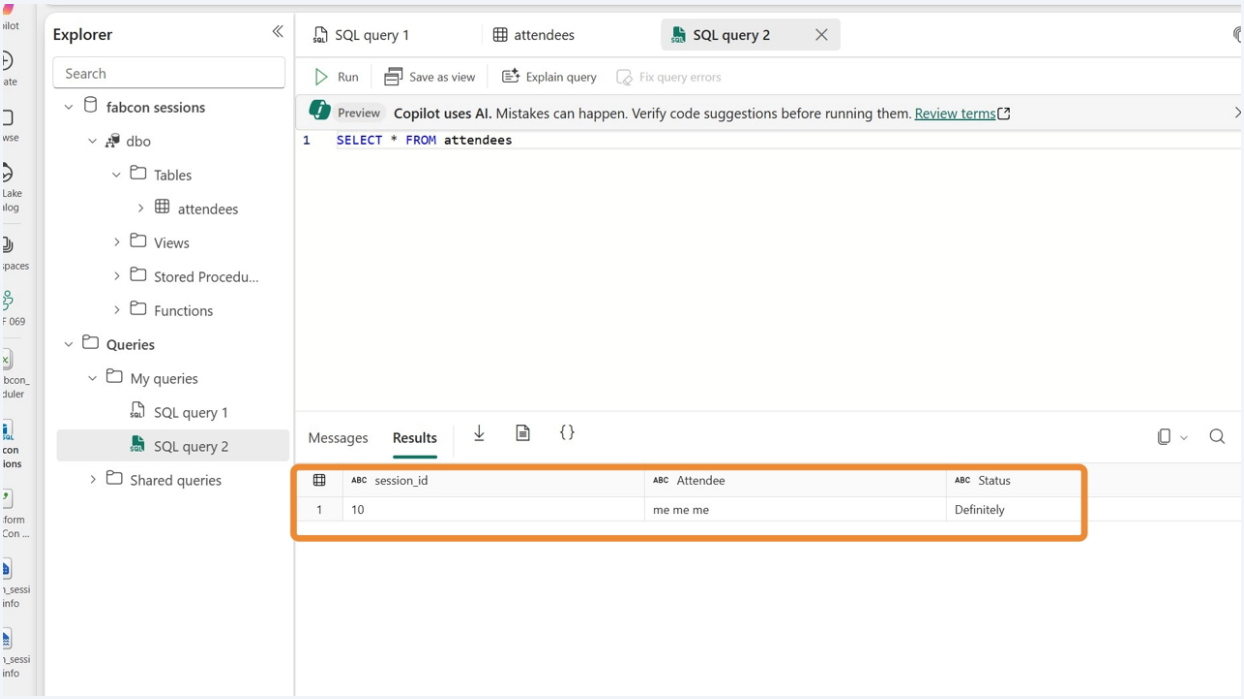
32 If the preview doesn't show anything yet, you can see it by writing a SQL query. Click "New Query."



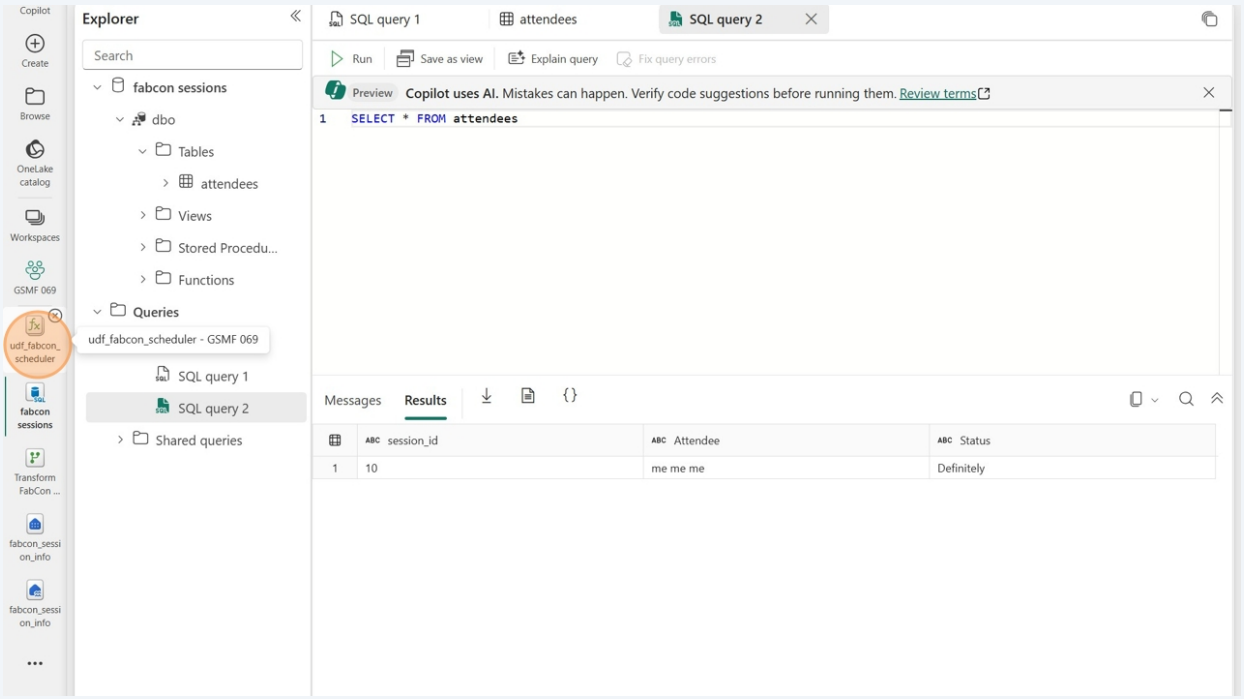
33 Enter "SELECT * FROM attendees" and then click the "Run" button.



34 You should now see one row. You populated the table by running the UDF!

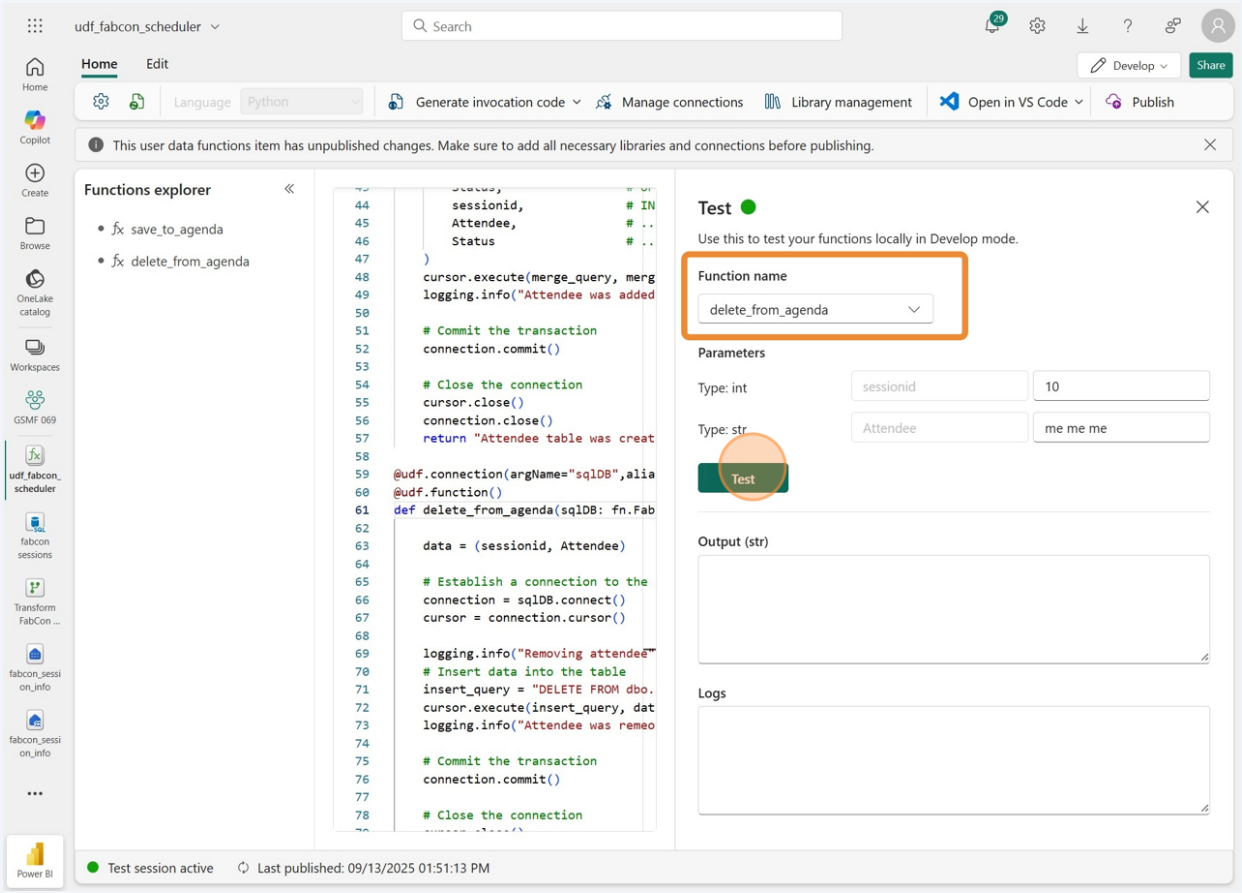


35 Now go back to the UDF so we can test the delete function. You'll need to use the same session_id and Attendee so it can delete the correct row.

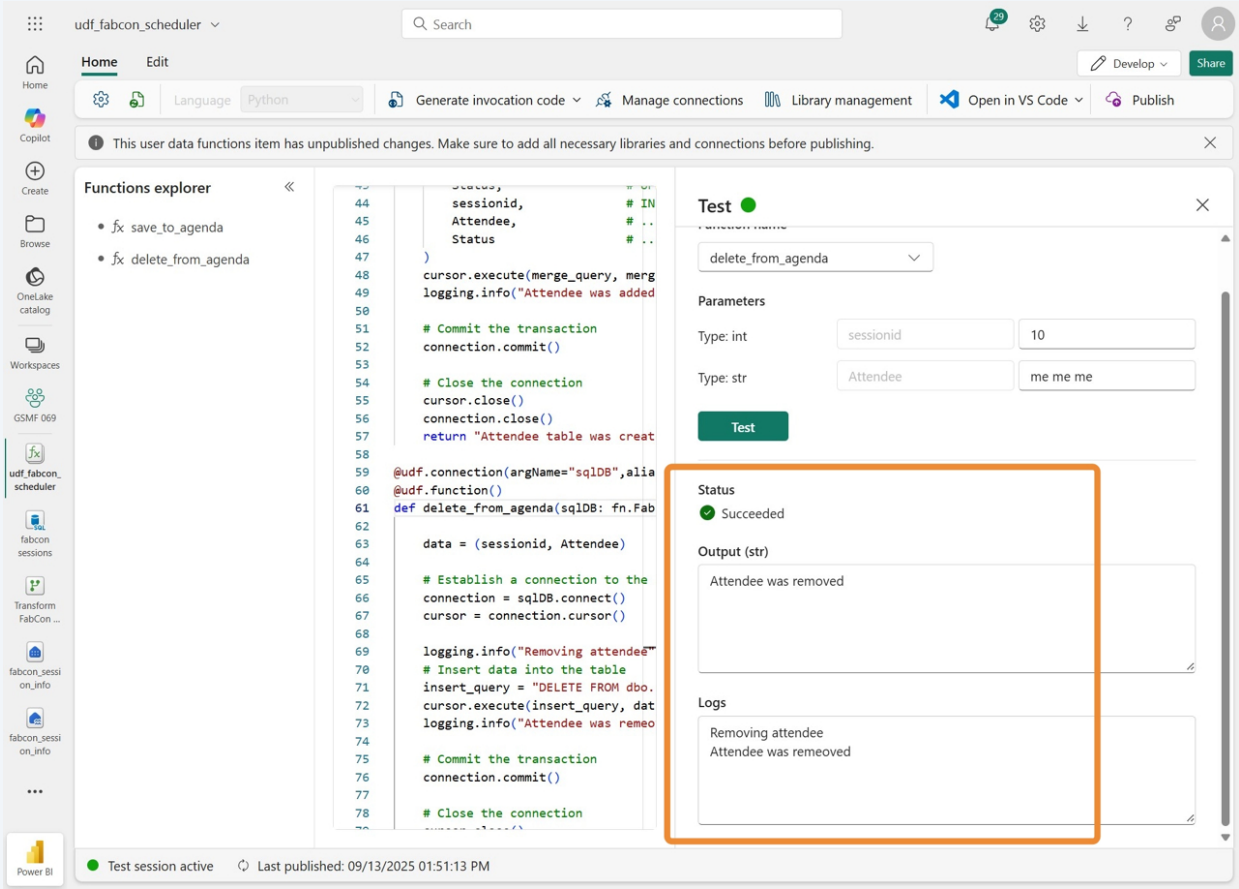


Test the second function

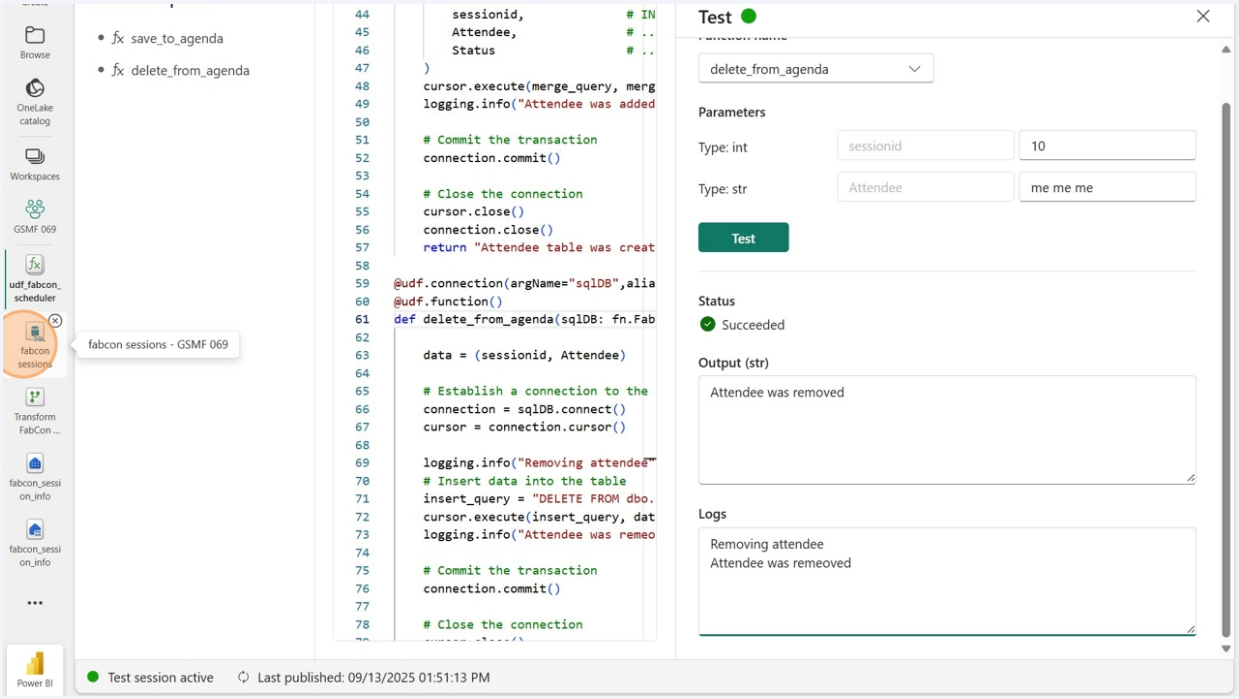
36 If you didn’t close the test window, you can simply scroll to the top and select the delete function in the Function name dropdown. This should leave in the parameters you already had. If you did close it, just hover over the delete function and click the test tube to test. Then fill in the same sessionid and Attendee values you used for the first test and click the “Test” button.



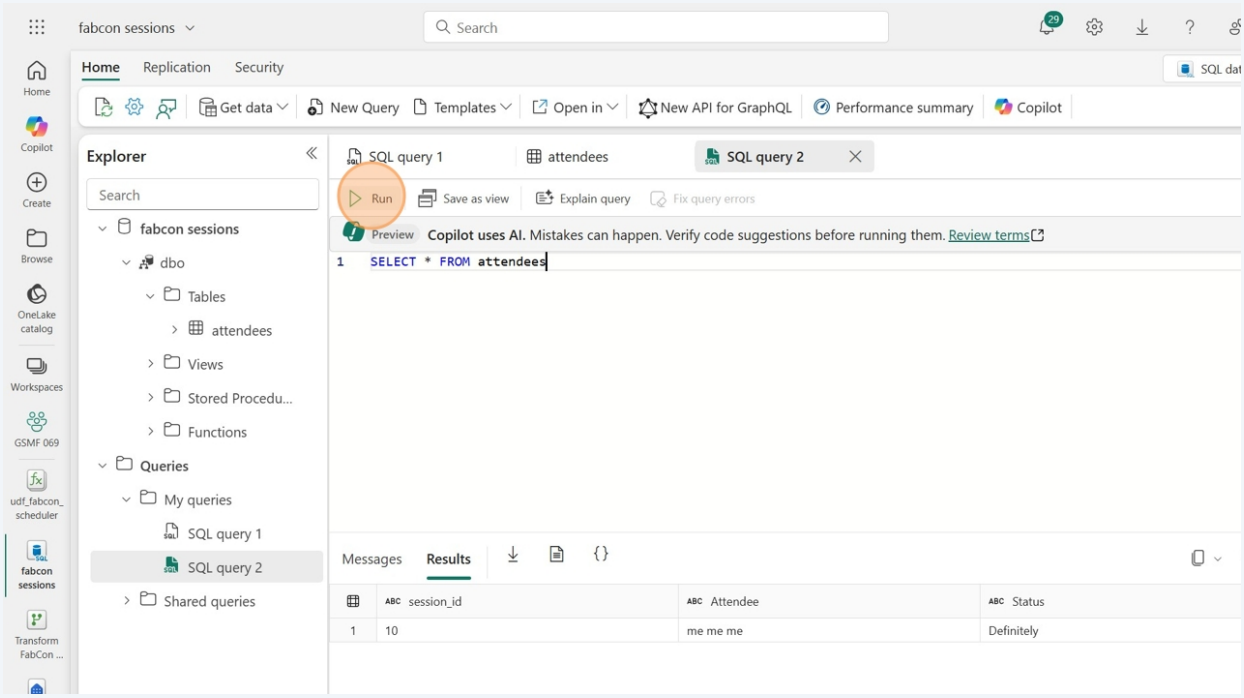
37 The output and logs should tell you that you deleted a row.



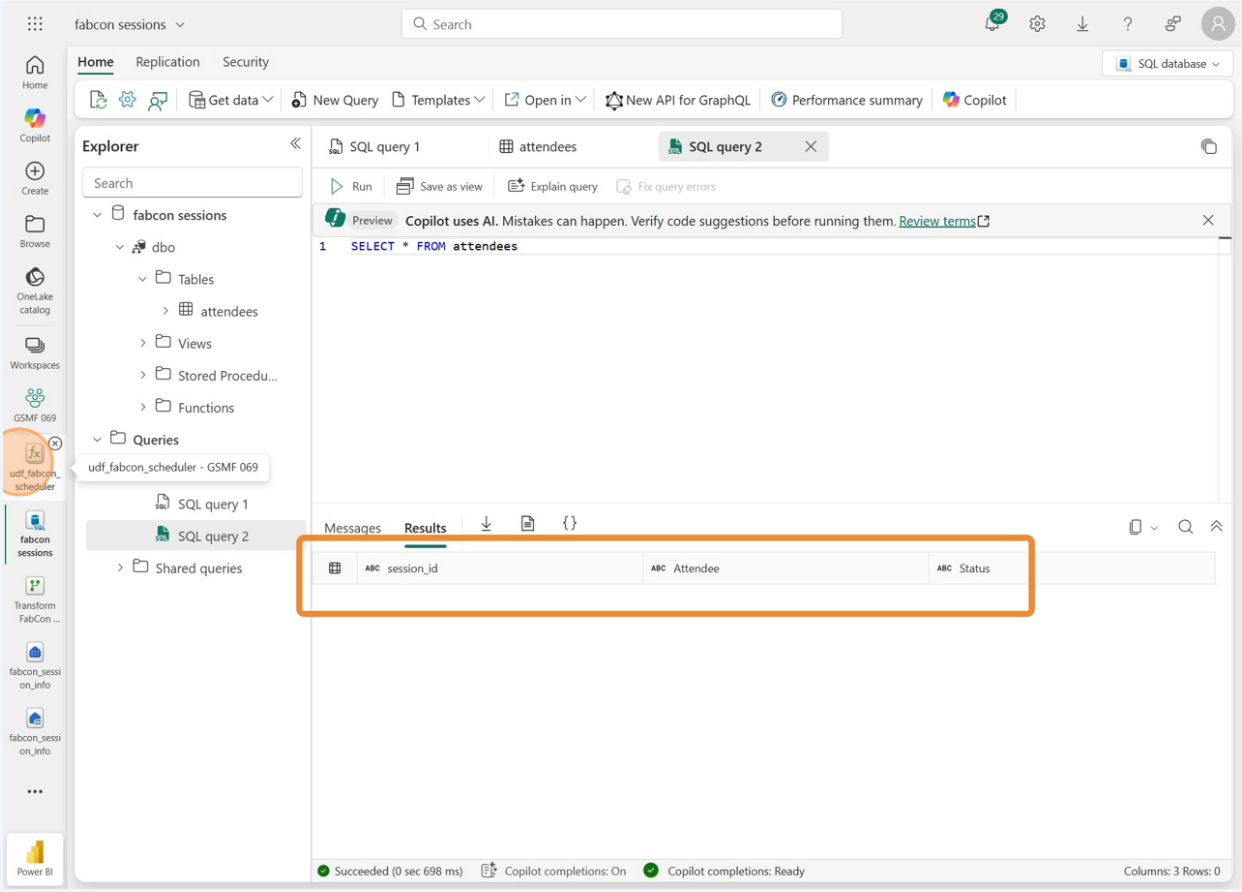
38 Go back to the SQL database again and we'll make sure the row is gone.



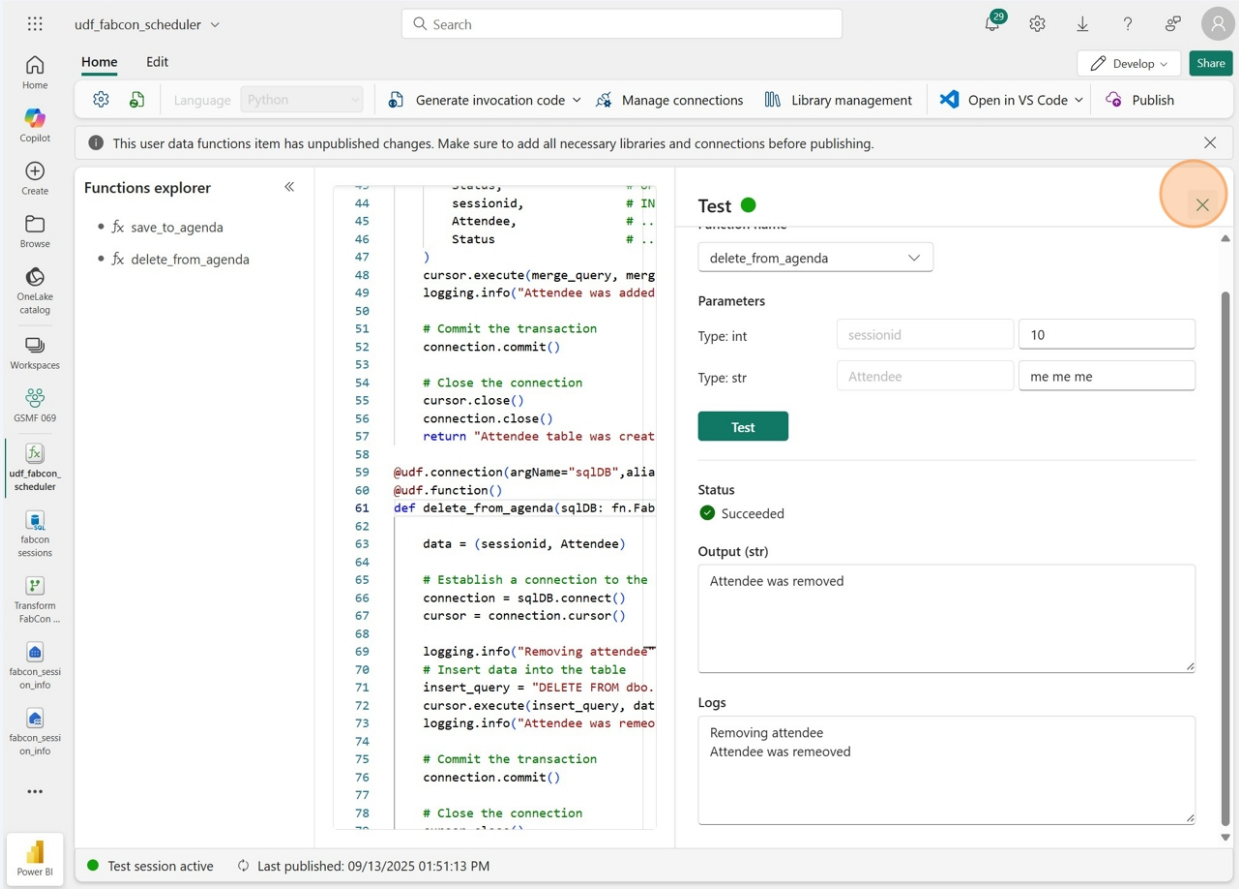
39 Rerun the SQL query.



40 The row is gone! Back to the empty table. Navigate back to the UDF.

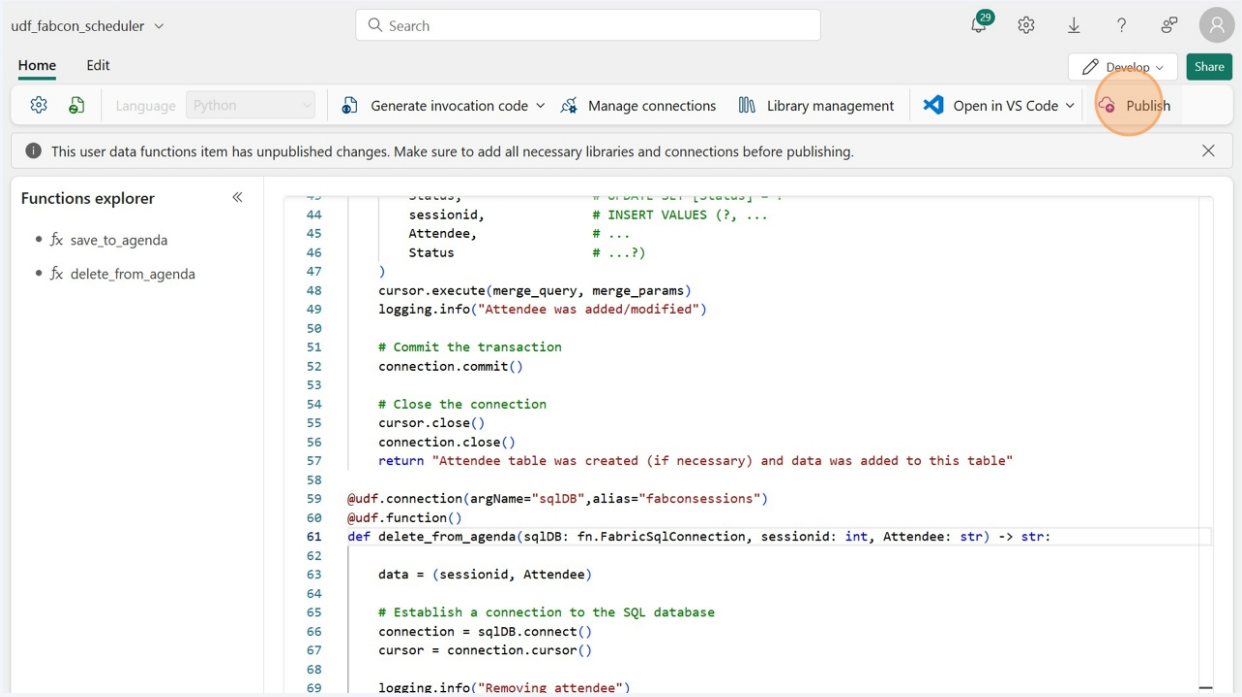


41 Close the test pane if you've still got it open.

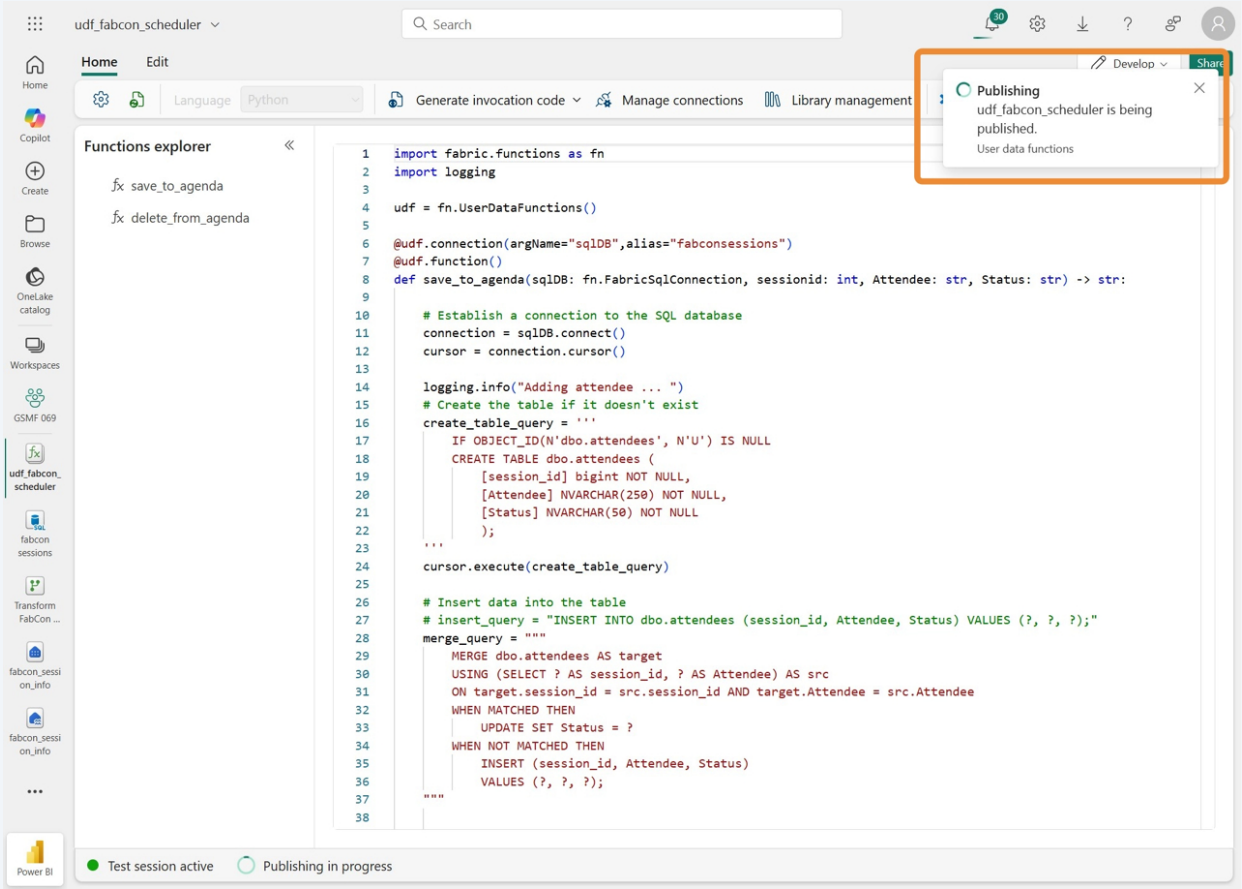


Publish the User data function

42 Click "Publish" to publish the function.



43 You'll see a message that the function is being published.



44 Go back to the workspace and you'll see that you now have a User Data Function.