Data to Fish

Menu

Import a CSV File to SQL Server using Python

October 1, 2021

There is more than one way to import a CSV file to SQL Server using *Python.* In this guide, you'll see a simple technique to import your data using the following 2 Python libraries:

- Pandas – used to import the CSV file into Python and create a DataFrame
- Pyodbc – used to connect Python to SQL Server

Steps to Import a CSV file to SQL Server using Python

Step 1: Prepare the CSV File

To begin, prepare the CSV file that you'd like to import to SQL Server.

For example, let's assume that a CSV file was prepared, where:

- The CSV file name is **'products'**
- The CSV file is stored under the following path: **C:\Users\Ron\Desktop\Test\products.csv**

In addition, the CSV file contains the following data:

product_id	product_name	price

1	Laptop	1200
2	Printer	200
3	Tablet	350
4	Keyboard	80
5	Monitor	400

Step 2: Import the CSV File into a DataFrame

You may use the *Pandas* library to import the CSV file into a DataFrame.

Here is the code to import the CSV file for our example (note that you'll need to change the path to reflect the location where the CSV file is stored on *your* computer):

```
import pandas as pd

data = pd.read_csv (r'C:\Users\Ron\Desktop\Test\products.csv')
df = pd.DataFrame(data)

print(df)
```

This is how the DataFrame would look like in Python:

```
   product_id  product_name  price
0           1        Laptop   1200
1           2       Printer    200
2           3        Tablet    350
3           4      Keyboard     80
4           5       Monitor    400
```

Step 3: Connect Python to SQL Server

To connect Python to SQL server, you'll need the:

- *Server Name*. For demonstration purposes, let’s assume that the server name is: **RON\SQLEXPRESS**
- *Database Name*. The database name for our example would be: **test_database**

Here is the code to connect Python to SQL for our example:

```
import pyodbc

conn = pyodbc.connect('Driver={SQL Server};'
                      'Server=RON\SQLEXPRESS;'
                      'Database=test_database;'
                      'Trusted_Connection=yes;')
cursor = conn.cursor()
```

You may wish to check the following guide that explains the full steps to connect Python to SQL Server using pyodbc.

Step 4: Create a Table in SQL Server using Python

Next, add the syntax to create the table in SQL Server. This table will be used to store the imported data from the CSV file.

For our example, you can add the following syntax to create the ‘**products**‘ table:

```
cursor.execute('''
                CREATE TABLE products (
                        product_id int primary key,
                        product_name nvarchar(50),
                        price int
                        )
               ''')
```

Note that whenever you run the code to create a table. You should only use that piece of the code once. Otherwise, you’ll get the error below:

> *ProgrammingError: (’42S01', “[42S01] [Microsoft][ODBC SQL Server Driver][SQL Server]There is already an object named ‘products’ in the database. (2714) (SQLExecDirectW)”)*

Step 5: Insert the DataFrame Data into the Table

Here is the syntax to insert the DataFrame data (from step-2) into the **products** table:

```
for row in df.itertuples():
    cursor.execute('''
                INSERT INTO products (product_id, product_name, price)
                VALUES (?,?,?)
                ''',
                row.product_id,
                row.product_name,
                row.price
                )
conn.commit()
```

And here is the *entire* code to import the CSV file into SQL Server using Python:

```
import pandas as pd
import pyodbc

# Import CSV
data = pd.read_csv (r'C:\Users\Ron\Desktop\Test\products.csv')
df = pd.DataFrame(data)

# Connect to SQL Server
conn = pyodbc.connect('Driver={SQL Server};'
                      'Server=RON\SQLEXPRESS;'
                      'Database=test_database;'
                      'Trusted_Connection=yes;')
```

```
cursor = conn.cursor()

# Create Table
cursor.execute('''
                CREATE TABLE products (
                        product_id int primary key,
                        product_name nvarchar(50),
                        price int
                        )
               ''')

# Insert DataFrame to Table
for row in df.itertuples():
    cursor.execute('''
                INSERT INTO products (product_id, product_name, price)
                VALUES (?,?,?)
                ''',
                row.product_id,
                row.product_name,
                row.price
                )
conn.commit()
```

Run the code in Python (after making the adjustment to the path where your CSV file is stored, as well as making the change to your database connection info).

Step 6: Perform a Test

Let’s run a simple query to check that the values from the CSV file got imported into SQL Server:

```
select * from products
```

Here is the result:

product_id	product_name	price
1	Laptop	1200
2	Printer	200

3	Tablet	350
4	Keyboard	80
5	Monitor	400

Python

‹ Fastest way to Convert Integers to Strings in Pandas DataFrame

› How to Check the Version of Numpy Installed

Tutorials

Python Tutorials

R Tutorials

Julia Tutorials

Batch Scripts

Recent Posts

How to Iterate over a List of Lists in Python

How to Iterate over a Dictionary in Python

DATA TO FISH

Privacy Policy - Cookie Policy - Terms of Service

Home » Python » Import a CSV File to SQL Server using Python