**How to Bulk Load Data into PostgreSQL with Python**

Bulk loading with the copy command from a CSV file is the fastest option to load a large table with Postgres. In fact, loading data from a flat file is the fastest option in any relational databases. When you have a large table and need to load it to another database, the fastest way is to unload it to a flat file and upload it to the database table.

Let’s first import all the necessary modules. The [psycopg2](http://initd.org/psycopg/docs/install.html) module is pretty much your best friends to connect to Postgres with Python.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

**Exporting Table to CSV**

Let’s export a table to a csv file. To export an entire table, you can use select \* on the target table. Panda’s read\_sql function will convert the query result into Pandas’ dataframe. To create a CSV file, you can use to\_csv on the dataframe.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]

**Importing CSV to Table**

The copy\_expert function will give you good option to copy a CSV file. The function takes the input file path, target table and connection details. Note that the column order should be the same between the CSV file and the table.

[cc lang="python" tab\_size="4" lines="-1"]

[/cc]