**How to Bulk Load Data with JDBC and Python**

Let’s do data bulk load by using JDBC and Python. The aim of this post is pretty much the same as the [previous one with ODBC](https://www.mydatahack.com/how-to-bulk-load-data-with-odbc-and-python/). We are going to export a table into a csv file and import the exported file into a table by using JDBC drivers and Python. To interact with JDBC drivers, you need to install the JayDeBeApi module.

For some reason, I could not get JayBeDeApi to work with Python 2.7. When I install it, I kept getting the class not found error (such as Error: java.lang.RuntimeException: Class com.mysql.jdbc.Driver not found) although I had the correct class name and jar file path. When I switched to Python 3 with JayBeDeApi3 installation, the problem disappeared. So, code here only works on Python 3. For Python 3, you need to install JayBeDeApi3 as below.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

pip install JayBeDeApi3

[/cc]

Let’s first import the required modules.

Exporting CSV file from Table

The function takes a select statement and connection parameters. You have to make sure to have the correct class name (case sensitive!) and the path to the JDBC jar file. The rest is pretty straight forward. We are using pandas function to convert the query results into a data frame and creating a csv file from it.

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

[/cc]

Here is the execution example with MySQL JDBC.

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

[/cc]

**Importing CSV file to Table**

The function takes a flat file upload statement and connection parameters. If you do not set autocommit, you need to execute the commit statement. Either way works fine.

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

[/cc]

Here are the execution examples with MySQL and Postgres. Each database has SQL syntax for this and you need to pass the statement to the function. MySQL uses the [LOAD DATA INFILE](https://dev.mysql.com/doc/refman/5.7/en/load-data.html) command while Postgres uses the [copy command](https://www.postgresql.org/docs/10/static/sql-copy.html).

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

[/cc]