

TASK BRIEF

Your team has been tasked with collecting metrics on a plethora of disparate shipping data. This task comes straight from the top, so it would be wise to give it your all. The data is contained in a number of different spreadsheets, each with its own competing schema. In order to interrogate the data, all of it has to be in the same place and in the same format. Currently, the shipping data exists in several places in several formats and is therefore impossible to query. To combine the spreadsheets, you need to write a python script to read through every row, extract the pertinent data, figure out how to combine it, munge it into the right format, and upload it to the database. Plenty of steps, but the resulting data will be much easier to query. Once the database contains all the data, you can pass it off to the analysis team to extract all the relevant metrics.

Part 1: Get the data

First, you need to get your hands on the relevant data. The shipping department has been kind enough to provide you with a repository containing all of their spreadsheets, as well as a copy of the SQLite database. First, fork and clone the repository from the *TASK RESOURCES* folder.

Part 2: Populate the database

Your task is to insert all of the data contained in the provided spreadsheets into the SQLite database. You will write a Python script which:

- Reads each row from the spreadsheets.
- Extracts the relevant data.
- Munges it into a format that fits the database schema.
- Inserts the data into the database.

Spreadsheet 0 is self-contained and can simply be inserted into the database, but spreadsheets 1 and 2 are dependent on one another. Spreadsheet 1 contains a single product per row, you will need to combine each row based on its shipping identifier, determine the quantity of goods in the shipment, and add a new row to

the database for each product in the shipment. The origin and destination for each shipment in spreadsheet 1 are contained in spreadsheet 2. You may assume that all the given data is valid – product names are always spelt the same way, quantities are positive, etc.

When you're finished, convert the python script you used to populate the database into a PDF and submit it.

ADDITIONAL RESOURCES

1. CSV File Reading And Writing – <https://docs.python.org/3/library/csv.html>

2. DB- API 2.0 Interface for SQLite Databases – <https://docs.python.org/3/library/sqlite3.html>