Data Engineer Technical Assessment

Use these tables for questions one and two:  
 **orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id (integer) | product\_id (integer) | ordered\_at (datetime) | product\_quantity (string) | created\_at (datetime) | updated\_at (datetime) |
| 1 | 3 | 2021-01-01 00:00:00 | 1 | 2021-01-01 00:00:01 | 2021-01-01 00:01:03 |
| 2 | 1 | 2021-01-03 05:00:07 | 4 | 2021-01-03 05:00:08 | 2021-01-03 05:02:12 |
| 3 | 5 | 2021-01-05 12:05:00 | 3 | 2021-01-05 12:05:01 | 2021-01-06 01:05:00 |
| 4 | 1 | 2021-02-01 03:00:00 | 2 | 2021-02-01 03:00:01 | 2021-02-01 03:00:01 |
| 5 | 1 | 2021-02-05 00:00:00 | 3 | 2021-02-05 00:00:01 | 2021-02-05 00:00:01 |

**products**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id (integer) | name  (string) | description (string) | go\_live\_date (date) | created\_at (datetime) | updated\_at  (datetime) |
| 1 | Mattress | A Mattress | 2021-01-01 | 2020-12-01 00:00:00 | 2020-12-01 00:00:00 |
| 3 | Pillow | A Pillow | 2021-01-01 | 2020-12-01 00:00:00 | 2020-12-01 00:00:00 |
| 5 | Comforter | A Comforter | 2021-01-01 | 2020-12-01 00:00:00 | 2020-12-01 00:00:00 |

**marketing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id  (integer) | ad\_network (string) | source  (string) | created\_at (datetime) | updated\_at (datetime) |
| 1 | facebook | ads | 2020-12-01 00:00:00 | 2020-12-01 00:00:00 |
| 2 | facebook | remarketing | 2020-12-01 00:00:00 | 2020-12-01 00:00:00 |

**marketing\_orders**

|  |  |
| --- | --- |
| order\_id  (integer) | marketing\_id  (integer) |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |
| 5 | 2 |

1. Using the raw tables above, write a query that returns the product name, quantity sold, marketing ad network, and marketing source for the best-selling product and most successful marketing source in the month of January 2021.
2. Using the raw tables above as initial input (staging tables), model the tables so that they support two downstream reports. This problem should have four parts in the response including: The SQL used to model the staging tables above, the SQL used to build order\_products from that table, the SQL used to build marketing\_orders also from the modeled tables, and finally a short explanation on why you built it as such.
   1. order\_products report: This table has these columns:
      1. id (a unique identifier)
      2. product\_id (id of product)
      3. num\_sold (number of products sold)
      4. month (a date of the first of the month for each month, e.g. 2021-01-01)
   2. marketing\_orders report: This table has these columns:
      1. id (a unique identifier)
      2. ad\_network (name of the ad network associated to the marketing row)
      3. source (source of the marketing row)
      4. top\_product (the top product sold under the ad network + source)
      5. top\_product\_quantity (quantity of the top product sold under the ad network and source)
      6. most\_recent\_sold\_date (most recent date that this top product was sold in the month)
3. Create a python class (or multiple if needed) that handles taking an excel file, converting it into a csv file, and then makes a get request to <https://www.google.com> and returns the status code. Finally, write tests to cover your code. Use whatever python packages you want.  
   XLSX Format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | name | description | commas\_used | created\_at | updated\_at |
| 1 | Test | Test Entry | ,,, | 2020-01-01 | 2020-01-01 |
| 2 | Test1 | Test 1 Entry | ,,,,,, | 2020-01-01 | 2020-10-05 |
| 3 | Test2 | Test 2 Entry | , | 2020-03-04 | 2020-03-05 |

CSV File Format:  
id (integer), name (string), commas\_count (integer, source: commas\_used, convert count of commas to a number), commas\_used (string), created\_at (datetime), updated\_at (date)

Deliverables:

* 1. Python class/classes to handle the process described above
  2. Tests to cover the class/classes
  3. Optional requirements file

1. Create a Dockerfile that installs the necessary packages to run the code from question #3. You may use whatever source you wish to do so.  
   Deliverable: Just the Dockerfile, no link to a built image is necessary but you should make sure that it does build properly.