Innovation Mindset

Cybersecurity Audit of Enron Emails

Overview

Enron Corporation was a very large energy, commodities, and service company that was based in Houston, Texas. The company became famous for an accounting fraud that was discovered in 2001. As part of the discovery of the fraud, federal investigators examined the emails sent by top company employees. They also posted the emails online so others could search through them. This email repository has become very popular for understanding how employees use email and for performing analysis on “real-world” data.

For this case, you will perform a few cybersecurity audit procedures on 26,751 emails sent by top Enron employees. The cybersecurity audits procedures you will perform are designed to test if employees are adhering to cybersecurity policies that have been (hypothetically) adopted by Enron Corporation.

The emails included in the dataset are real, unless your professor altered and/or added/deleted a few emails for learning purposes. This email corpus does not include all emails released; it has been limited in several ways. For example, group emails are excluded and other emails that required advanced data analytic procedures are excluded. The email data has been modified to simplify some of the tasks you are asked to perform.

The file, “Email File parsed.CSV” contains the data in a CSV file. Each email is listed on its own line.

The cybersecurity audit procedures should be performed using Alteryx software. Alteryx software is a strong candidate for this type of analysis because it can be easily applied to future emails by changing the input file. Also, Alteryx allows for powerful manipulation that is auditable.

Suggestions

Please consider the following suggestions:

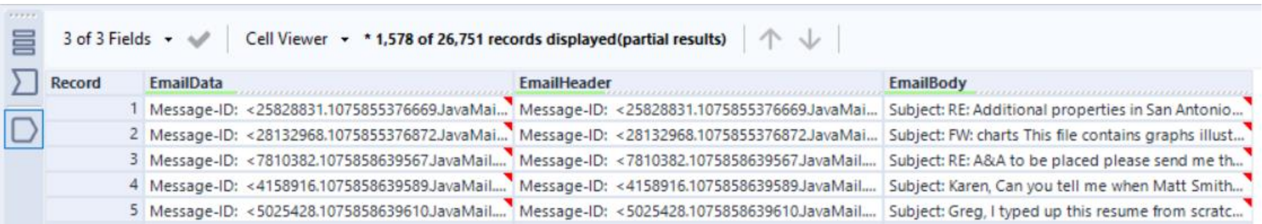
- There are multiple ways to answer each question. You will be graded on correctly answering the question, not on how you answered the question. That said, succinct answers are preferred.
- Use the Comment Tool and Annotations to label what the code is designed to do. In practice, this helps others understand your logic and intention. In an educational setting, it allows the teacher to correctly grade your work.
- For any pattern matching and data parsing, you must use regular expressions (the RegEx Tool) to perform the activity. You can choose to combine multiple RegEx expressions into a single instance of the tool or use the tool multiple times to extract information.
- All answers should be included in a single Alteryx file (.yxmd file type). This can be done by answering question 1 and then branching from the end of your sequence that answers question 1 to answer each additional question. That is, questions 2-6 build off the solution for question 1, but do not build off each other. There is a screenshot at the end of the case that provides an example of what this could look like.

- For purposes of this assignment, all RegEx expressions should _not_ be case sensitive.
- Be aware that finding emails that match on certain criteria does not indicate the Enron employee necessarily violated the cybersecurity control. You will need to manually review the results that are flagged by your searches and use your professional judgment to decide if the email violates the cybersecurity policy or not (or if additional testing would be warranted or not).

Required

Your assignment is to do the following in Alteryx.

1. Prepare the file for analysis. To do this, do the following:
   a. Split the header information from the body of the message. The header data is all data that comes before the subject line in an email. That is, the header contains the message id, the email of the message sender, and the email address of the message receiver. The body of the message contains the subject line and all information until the end of the email. Label the email header as "EmailHeader" and the body of the message as "EmailBody". The screenshot shows you what the output should look like. All screenshots show only a few rows and not necessarily all answers.



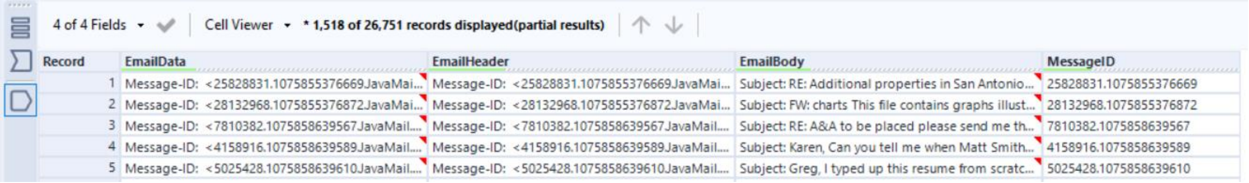
3 of 3 Fields | Cell Viewer | * 1,578 of 26,751 records displayed(partial results)

Record	EmailData	EmailHeader	EmailBody
1	Message-ID: <25828831.1075855376669.JavaMai...	Message-ID: <25828831.1075855376669.JavaMai...	Subject: RE: Additional properties in San Antonio...
2	Message-ID: <28132968.1075855376872.JavaMai...	Message-ID: <28132968.1075855376872.JavaMai...	Subject: FW: charts This file contains graphs illust...
3	Message-ID: <7810382.1075858639567.JavaMail....	Message-ID: <7810382.1075858639567.JavaMail....	Subject: RE: A&A to be placed please send me th...
4	Message-ID: <4158916.1075858639589.JavaMail....	Message-ID: <4158916.1075858639589.JavaMail....	Subject: Karen, Can you tell me when Matt Smith...
5	Message-ID: <5025428.1075858639610.JavaMail....	Message-ID: <5025428.1075858639610.JavaMail....	Subject: Greg, I typed up this resume from scratc...

   A few hints might be helpful:

   - Make sure to increase the field length when you import the data or else data will be truncated on the import.
   - For your regular expression, use a "non-greedy" operator with a quantifier. Non-greedy operators stop searching once it finds the first match for the term searching from the beginning of the string to the end of the string, whereas greedy operators search from the end of the string to the beginning of the string until it finds the matching term. You need to do this because some emails contain the word "Subject:" multiple times (e.g., forwarded emails often do this). To make a quantifier non-greed add the ? after the quantifier. For example, "+?", makes the + quantifier non-greedy.

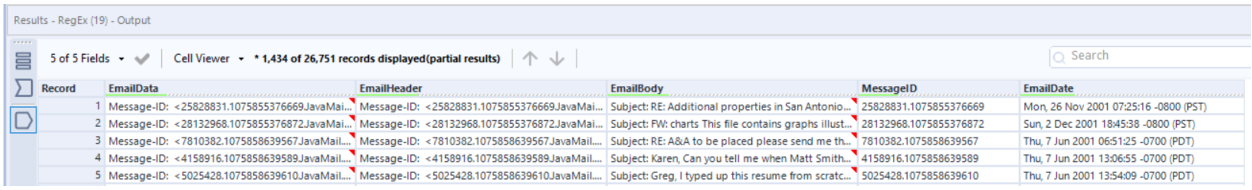
   b. From the message header, extract the Message-ID. Extract just the digits from the header. For example, a Message-ID might look like the following: <25828831.1075855376669.JavaMail.evans@thyme>, you should just extract 25828831.1075855376669 as the Message-ID. Label the extracted data as "MessageID". The screenshot shows you what the output should look like.



4 of 4 Fields ▾ Cell Viewer ▾ * 1,518 of 26,751 records displayed(partial results)

Record	EmailData	EmailHeader	EmailBody	MessageID
1	Message-ID: <25828831.1075855376669.JavaMai...	Message-ID: <25828831.1075855376669.JavaMai...	Subject: RE: Additional properties in San Antonio...	25828831.1075855376669
2	Message-ID: <28132968.1075855376872.JavaMai...	Message-ID: <28132968.1075855376872.JavaMai...	Subject: FW: charts This file contains graphs illust...	28132968.1075855376872
3	Message-ID: <7810382.1075858639567.JavaMail....	Message-ID: <7810382.1075858639567.JavaMail....	Subject: RE: A&A to be placed please send me th...	7810382.1075858639567
4	Message-ID: <4158916.1075858639589.JavaMail....	Message-ID: <4158916.1075858639589.JavaMail....	Subject: Karen, Can you tell me when Matt Smith...	4158916.1075858639589
5	Message-ID: <5025428.1075858639610.JavaMail....	Message-ID: <5025428.1075858639610.JavaMail....	Subject: Greg, I typed up this resume from scratc...	5025428.1075858639610

c. From the message header, extract the date. The date information will appear like “Sun, 2 Dec 2001 18:45:38 -0800 (PST)”. Label this data as “EmailDate”. The screenshot shows you what the output should look like.

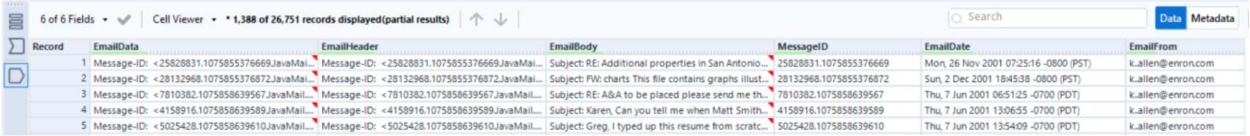


Results - RegEx (19) - Output

5 of 5 Fields ▾ Cell Viewer ▾ * 1,434 of 26,751 records displayed(partial results)

Record	EmailData	EmailHeader	EmailBody	MessageID	EmailDate
1	Message-ID: <25828831.1075855376669.JavaMai...	Message-ID: <25828831.1075855376669.JavaMai...	Subject: RE: Additional properties in San Antonio...	25828831.1075855376669	Mon, 26 Nov 2001 07:25:16 -0800 (PST)
2	Message-ID: <28132968.1075855376872.JavaMai...	Message-ID: <28132968.1075855376872.JavaMai...	Subject: FW: charts This file contains graphs illust...	28132968.1075855376872	Sun, 2 Dec 2001 18:45:38 -0800 (PST)
3	Message-ID: <7810382.1075858639567.JavaMail....	Message-ID: <7810382.1075858639567.JavaMail....	Subject: RE: A&A to be placed please send me th...	7810382.1075858639567	Thu, 7 Jun 2001 06:51:25 -0700 (PDT)
4	Message-ID: <4158916.1075858639589.JavaMail....	Message-ID: <4158916.1075858639589.JavaMail....	Subject: Karen, Can you tell me when Matt Smith...	4158916.1075858639589	Thu, 7 Jun 2001 13:06:55 -0700 (PDT)
5	Message-ID: <5025428.1075858639610.JavaMail....	Message-ID: <5025428.1075858639610.JavaMail....	Subject: Greg, I typed up this resume from scratc...	5025428.1075858639610	Thu, 7 Jun 2001 13:54:09 -0700 (PDT)

d. From the message header, extract the email address of the person who sent the email. Label this data as “EmailFrom”. The screenshot shows you what the output should look like.

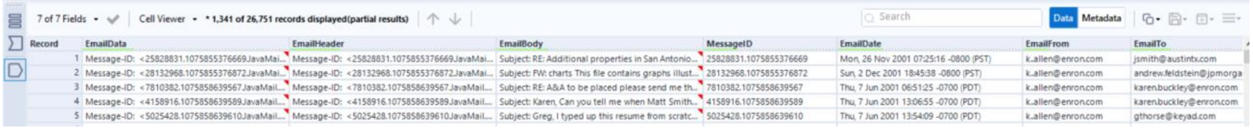


6 of 6 Fields ▾ Cell Viewer ▾ * 1,388 of 26,751 records displayed(partial results)

Record	EmailData	EmailHeader	EmailBody	MessageID	EmailDate	EmailFrom
1	Message-ID: <25828831.1075855376669.JavaMai...	Message-ID: <25828831.1075855376669.JavaMai...	Subject: RE: Additional properties in San Antonio...	25828831.1075855376669	Mon, 26 Nov 2001 07:25:16 -0800 (PST)	k.allen@enron.com
2	Message-ID: <28132968.1075855376872.JavaMai...	Message-ID: <28132968.1075855376872.JavaMai...	Subject: FW: charts This file contains graphs illust...	28132968.1075855376872	Sun, 2 Dec 2001 18:45:38 -0800 (PST)	k.allen@enron.com
3	Message-ID: <7810382.1075858639567.JavaMail....	Message-ID: <7810382.1075858639567.JavaMail....	Subject: RE: A&A to be placed please send me th...	7810382.1075858639567	Thu, 7 Jun 2001 06:51:25 -0700 (PDT)	k.allen@enron.com
4	Message-ID: <4158916.1075858639589.JavaMail....	Message-ID: <4158916.1075858639589.JavaMail....	Subject: Karen, Can you tell me when Matt Smith...	4158916.1075858639589	Thu, 7 Jun 2001 13:06:55 -0700 (PDT)	k.allen@enron.com
5	Message-ID: <5025428.1075858639610.JavaMail....	Message-ID: <5025428.1075858639610.JavaMail....	Subject: Greg, I typed up this resume from scratc...	5025428.1075858639610	Thu, 7 Jun 2001 13:54:09 -0700 (PDT)	k.allen@enron.com

As a hint, some of the emails have been altered in the dataset. For this problem, if you return a result that looks like “legal <.taylor@enron.com>" as part of your response that is ok. You do not have to parse out the “legal (.” portion of the email.

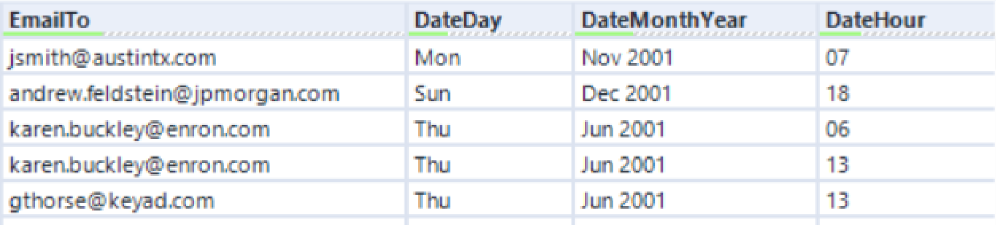
e. From the message header, extract the email address of the person who received the email. Label this data as “EmailTo”. The screenshot shows you what the output should look like.



7 of 7 Fields ▾ Cell Viewer ▾ * 1,341 of 26,751 records displayed(partial results)

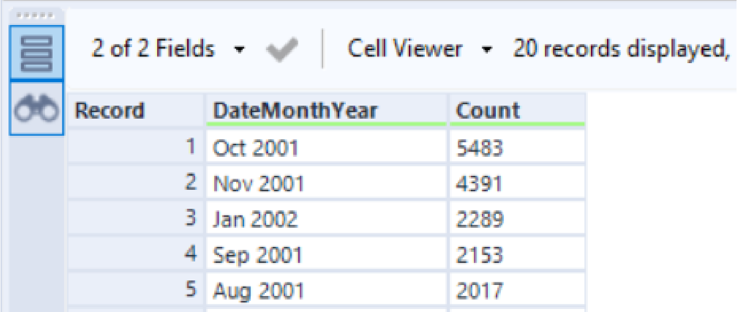
Record	EmailData	EmailHeader	EmailBody	MessageID	EmailDate	EmailFrom	EmailTo
1	Message-ID: <25828831.1075855376669.JavaMai...	Message-ID: <25828831.1075855376669.JavaMai...	Subject: RE: Additional properties in San Antonio...	25828831.1075855376669	Mon, 26 Nov 2001 07:25:16 -0800 (PST)	k.allen@enron.com	jsmith@austintx.com
2	Message-ID: <28132968.1075855376872.JavaMai...	Message-ID: <28132968.1075855376872.JavaMai...	Subject: FW: charts This file contains graphs illust...	28132968.1075855376872	Sun, 2 Dec 2001 18:45:38 -0800 (PST)	k.allen@enron.com	andrew.feldstein@jpmorga
3	Message-ID: <7810382.1075858639567.JavaMail....	Message-ID: <7810382.1075858639567.JavaMail....	Subject: RE: A&A to be placed please send me th...	7810382.1075858639567	Thu, 7 Jun 2001 06:51:25 -0700 (PDT)	k.allen@enron.com	karen.buckley@enron.com
4	Message-ID: <4158916.1075858639589.JavaMail....	Message-ID: <4158916.1075858639589.JavaMail....	Subject: Karen, Can you tell me when Matt Smith...	4158916.1075858639589	Thu, 7 Jun 2001 13:06:55 -0700 (PDT)	k.allen@enron.com	karen.buckley@enron.com
5	Message-ID: <5025428.1075858639610.JavaMail....	Message-ID: <5025428.1075858639610.JavaMail....	Subject: Greg, I typed up this resume from scratc...	5025428.1075858639610	Thu, 7 Jun 2001 13:54:09 -0700 (PDT)	k.allen@enron.com	gthorse@keyad.com

2. To gain an understanding of emailing behavior, output each of the following items. For this problem, output the requested item using a Browse activity. As a hint, you will need to parse the “EmailDate” field that you extracted in problem 1c above into the following fields “DateDay” (shows just the three letter code for day), “DateMonthYear” (which shows the 3 letter month and four digit year), and “DateHour” (which shows the two digit hour). The screenshot shows what the output from this step would look like (the screenshot does not show all columns or rows of data).



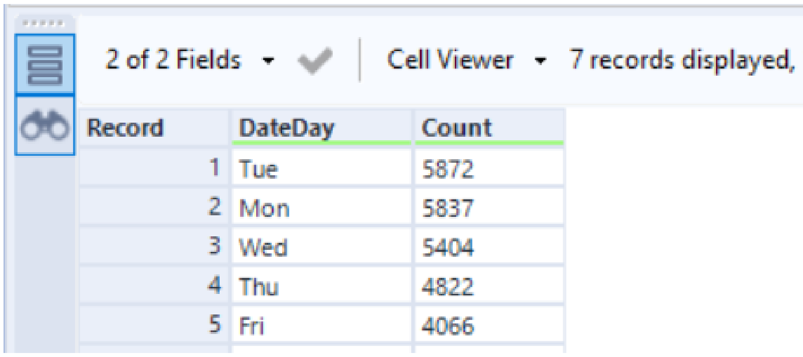
EmailTo	DateDay	DateMonthYear	DateHour
jsmith@austintx.com	Mon	Nov 2001	07
andrew.feldstein@jpmorgan.com	Sun	Dec 2001	18
karen.buckley@enron.com	Thu	Jun 2001	06
karen.buckley@enron.com	Thu	Jun 2001	13
gthorse@keyad.com	Thu	Jun 2001	13

a. Output the number of emails that are sent each month for each year. Sort the table so that the month/year that has the most emails is listed at the top. The screenshot shows you what the output should look like.



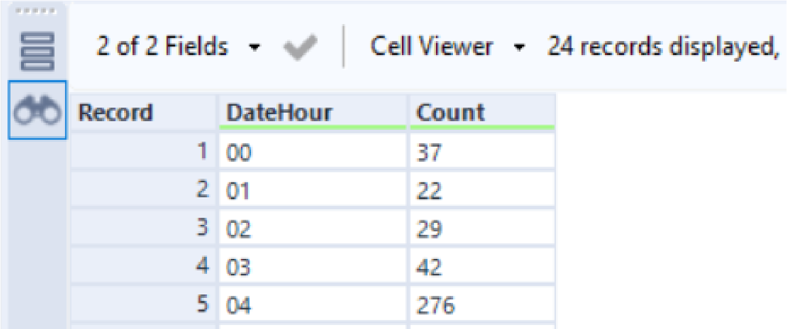
Record	DateMonthYear	Count
1	Oct 2001	5483
2	Nov 2001	4391
3	Jan 2002	2289
4	Sep 2001	2153
5	Aug 2001	2017

b. Output the number of emails that are sent each day of the week. You do not have to worry about different time zones for this analysis. You can use the hour of the day for the time zone they are in (i.e., you can combine hours without considering time zone). Sort the table so that the day of the week that has the most emails is listed at the top. The screenshot shows you what the output should look like.



Record	DateDay	Count
1	Tue	5872
2	Mon	5837
3	Wed	5404
4	Thu	4822
5	Fri	4066

c. Output the number of emails that are sent each hour of the day. Sort the table so that the hour of the day is shown in ascending order. The screenshot shows you what the output should look like.



Record	DateHour	Count
1	00	37
2	01	22
3	02	29
4	03	42
5	04	276

3. Company policy says that employees should not send sensitive information via email unless the email is encrypted. Test to see if the following sensitive information was sent in the body of an unencrypted email (note all emails in the file are unencrypted): social security numbers (SSN), employer identification numbers (EIN), and the individual taxpayer identification number (ITIN). You are only required to search for these numbers that follow the patterns shown below.

Be aware that the RegEx “Match” function in Alteryx requires a match of the *entire string* and not just part of the string. Thus, your expressions will need to match the entire body of the email when searching for these numbers (as a hint, include a “.*” at the beginning and end of each search string).

a. SSN: “###-##-####” (the search should include the dashes).

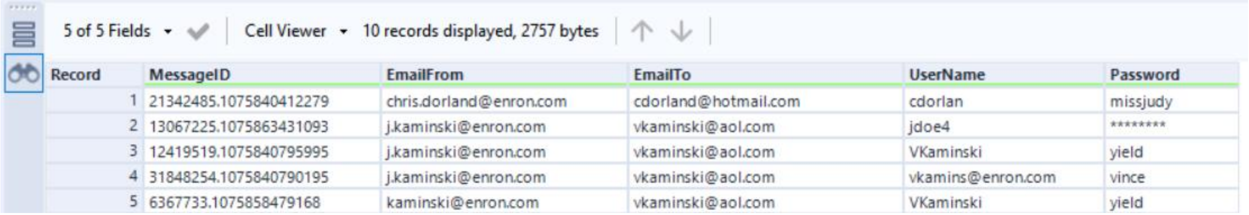
b. EIN: “##-#######” (the search should include the dashes).

c. ITIN: “9##-##-####” (the search should include the 9 to begin and the dashes).

Combine your results from all three searches into a single list and display the output using a Browse activity. Display the following: (1) the Message-ID, (2) the email address of the person who sent the email, (3) the email address of the person who received the email, and (4) what information was improperly sent (email body) and (5) the pattern type.

Review each email and draw conclusions about whether the email is a violation of policy. For example, can you draw any inferences about what is happening from some of these scenarios? Which one seems most problematic? If the company were considering internal control changes for the future, what could they do to prevent these situations?

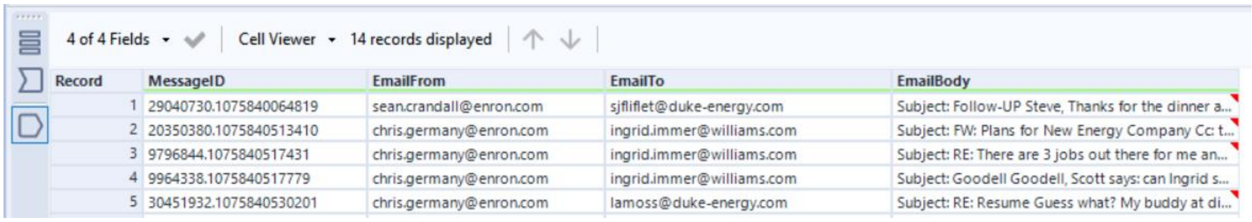
4. Company policy forbids the sharing of usernames and passwords. In the past, some employees have sent usernames and/or passwords to others via email. Search the email corpus for any email that may list usernames and passwords. To simplify this search, you should only find emails that include “username:” followed by “password:” where any text can be between the two words. Also, you should assume the word that immediately following “username:” is the username and the word immediately following “password:” is the password. Output using a Browse activity only the following: (1) the Message-ID, (2) the email address of the person who sent the email, (3) the email address of the person who received the email, (4) the username and (5) the password for those emails that do not have a null username. The screenshot shows you what the output should look like.



Record	MessageID	EmailFrom	EmailTo	UserName	Password
1	21342485.1075840412279	chris.dorland@enron.com	cdorland@hotmail.com	cdorlan	missjudy
2	13067225.1075863431093	j.kaminski@enron.com	vkaminski@aol.com	jdoe4	********
3	12419519.1075840795995	j.kaminski@enron.com	vkaminski@aol.com	VKaminski	yield
4	31848254.1075840790195	j.kaminski@enron.com	vkaminski@aol.com	vkamins@enron.com	vince
5	6367733.1075858479168	kaminski@enron.com	vkaminski@aol.com	VKaminski	yield

Review each email and draw conclusions about whether the email is a violation of policy. Provide recommendations of what the company should do moving forward.

5. Company policy prohibits senior level employees from interviewing with company competitors. All the employees for which you have emails are required to follow this policy. Test to see if any employees are discussing interviewing with competing firms. To perform this test, you should identify competitors. To identify competitors, search the email addresses of receivers for any of the following domain names (the domain name is the part of the email address after the @ symbol): “williams.com”, “dynegy.com”, “duke-energy.com”, “entergykoch.com”, “entergy.com”, “constellation.com”, “constellationmgt.com”, or “estutenws11.energy.williams.com”. Among emails sent to competitors, search for any emails that contain any of the following words, “resume”, “interview”, or “application”. Output your results using a Browse activity and display (1) the Message-ID, (2) the email address of the person who sent the email, (3) the email address of the person who received the email, and (4) the full text of the email. The screenshot shows you what the output should look like.



4 of 4 Fields ▾ ✓ | Cell Viewer ▾ 14 records displayed

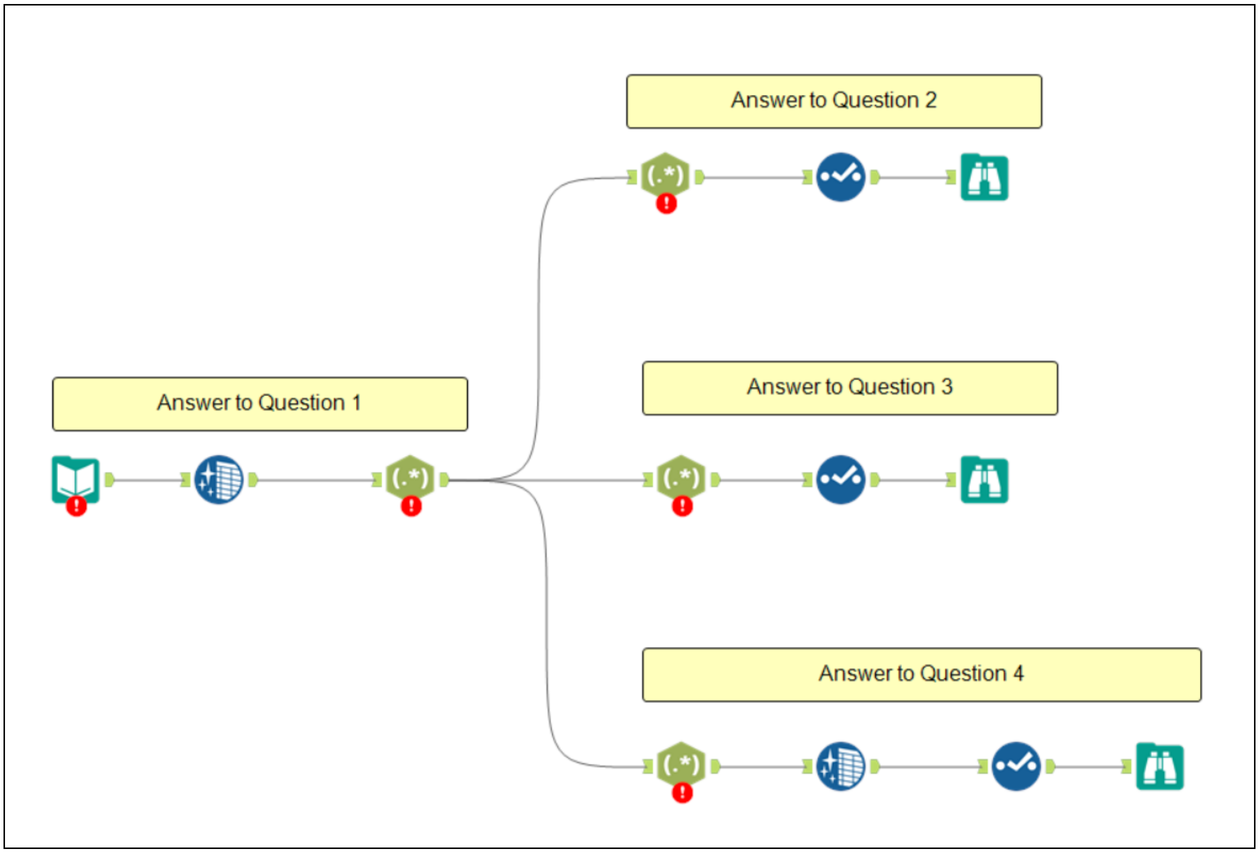
Record	MessageID	EmailFrom	EmailTo	EmailBody
1	29040730.1075840064819	sean.crandall@enron.com	sjfliflet@duke-energy.com	Subject: Follow-UP Steve, Thanks for the dinner a...
2	20350380.1075840513410	chris.germany@enron.com	ingrid.immer@williams.com	Subject: FW: Plans for New Energy Company Cc: t...
3	9796844.1075840517431	chris.germany@enron.com	ingrid.immer@williams.com	Subject: RE: There are 3 jobs out there for me an...
4	9964338.1075840517779	chris.germany@enron.com	ingrid.immer@williams.com	Subject: Goodell Goodell, Scott says: can Ingrid s...
5	30451932.1075840530201	chris.germany@enron.com	lamoss@duke-energy.com	Subject: RE: Resume Guess what? My buddy at di...

Read all of the emails and decide whether the employee violated company policy. Consider how you might improve your search string to better match emails that evidence violation of company policy and not flag emails that did not violate policy (you don't have to improve the string, but brainstrom ways that would improve the string).

6. Often employees can think of unique searches that reveal interesting and useful information. Write down a question that you would like to test and then answer it. Describe why you want to test your question and the findings from your testing.

Submit the following for grading:

- A Word file that lists the answers to the above questions.
- The single Alteryx file that performs all the above analyses. Use the "Comment Tool" to label what parts of your code answers each of the above questions. For example, the screenshot below shows how you might build this file (the activities are random in this screenshot).


Answer to Question 2
Answer to Question 1
Answer to Question 3
Answer to Question 4