Scenario

Your big data consulting company has been hired by a small law firm to help them make sense of a document dump they have received for a big trial.

The firm believes that the outcome of their trial depends on finding certain information in the emails from the opposition's clients.

They have secured an initial dump of employee's emails at the company in question, but in order to get continuing data they need to prove that there is value in the sample. In order for their document analysts to do that in a timely manner, they will need some metadata extracted from each email so they can process it using their document review tools.

If they are able to find what they need by the deadline, your company will get an ongoing contract to build a pipeline to process incoming document dumps (YAY!)

Assignment

Using the sample data consisting of a series of emails, write a Spark application to extract the Message ID, Date, From and To fields from each message's header, and output those fields along with the email contents into a CSV file.

Input:
Sample: https://bigdata220w18.blob.core.windows.net/blobs/enron_2015_sample.tgz
Full Set: https://bigdata220w18.blob.core.windows.net/blobs/enron_mail_20150507.tar.gz
The data is a tar gzipped file. Once expanded, the emails are in a dir structure of the following:

maildir/user/outlook-folder/message

Where each message is a numbered text file (1., 2., 3., etc..).

Output:

The output file should have the following format:

Message-ID,Date,From,To,Message

- Note that the email will most likely contain commas, so you will need to delimit the fields in the CSV file (" is the standard for this)
- All data should be included "as-is"; you don't need to do any further cleaning, i.e. Parse the date into a timestamp
- Also because of the newlines in the email, the last column will be very ugly, that is expected.

Example

Given the following email text:

```
Message-ID: <16159836.1075855377439.JavaMail.evans@thyme>
Date: Fri, 7 Dec 2001 10:06:42 -0800 (PST)
From: heather.dunton@enron.com
To: k..allen@enron.com
Subject: RE: West Position
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Dunton, Heather </O=ENRON/OU=NA/CN=RECIPIENTS/CN=HDUNTON>
X-To: Allen, Phillip K. </O=ENRON/OU=NA/CN=RECIPIENTS/CN=Pallen>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\Inbox
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst


Please let me know if you still need Curve Shift.

Thanks,
Heather
```

The corresponding line in the output CSV file would look like: (colored and split onto multiple lines for clarity)

```
<16159836.1075855377439.JavaMail.evans@thyme>,
"Fri, 7 Dec 2001 10:06:42 -0800 (PST)",
heather.dunton@enron.com,
k..allen@enron.com,
"Message-ID: <16159836.1075855377439.JavaMail.evans@thyme>
Date: Fri, 7 Dec 2001 10:06:42 -0800 (PST)
From: heather.dunton@enron.com
To: k..allen@enron.com
Subject: RE: West Position
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Dunton, Heather </O=ENRON/OU=NA/CN=RECIPIENTS/CN=HDUNTON>
X-To: Allen, Phillip K. </O=ENRON/OU=NA/CN=RECIPIENTS/CN=Pallen>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\Inbox
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst

Please let me know if you still need Curve Shift.

Thanks,
Heather"
```

Bonus Exercise

As an additional optional exercise, instead of a CSV file, write the output to a Hive table stored in HDFS. The only requirement for this would be to include the same 5 fields as in the CSV.

An example of a valid Hive table would look something like the following:

```
CREATE EXTERNAL TABLE emails(
        messageID STRING,
        dates STRING,
        fromAddr STRING,
        toAddr STRING,
        message STRING)
    COMMENT 'Email data dump'
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ','
    STORED AS ORC;
```