

Group members:

Anthony Tabet 243222

Elias Antoine Dargham 243184

Perla Saikaly 243228

Rim Kfoury 243492

Graph theory and Operational Research

Project Report

Abstract

This project report presents a comprehensive study of the Enron scandal using graph databases, focusing on the infamous Enron dataset - a collection of internal communications and financial data from the defunct Enron Corporation. The analysis hinges on the utilization of Neo4j, a graph database management system, which provides a nuanced platform for dissecting the complex network of email communications within Enron Corporation. By converting the dataset into a graph structure, our team was able to identify and analyze the intricate relationships among Enron employees, uncovering patterns indicative of fraudulent activities. Our methodology involved meticulously cleaning and preparing the Enron dataset for integration into Neo4j. We employed Python scripting for data preprocessing, ensuring the extraction of relevant information from email communications. This process facilitated the creation of a comprehensive graph representation of the dataset, with nodes representing employees and email messages and edges denoting email interactions. The crux of our analysis involved the formulation and execution of targeted queries within the Neo4j environment, aimed at unraveling the network of communication related to insider trading and other fraudulent practices. Our findings shed light on pivotal figures in the scandal, including Lou Pai and Andrew Fastow, and their roles in the company's downfall. The queries revealed a web of deceptive practices, including insider trading and the manipulation of financial statements through the LJM affair. The study concludes that the graph database approach offers a powerful lens through which the dynamics of corporate communication and malpractice can be examined. Our findings illustrate the role of key individuals in Enron's unethical practices, highlighting the efficacy of graph databases in analyzing large-scale, complex datasets. This project not only contributes to a deeper understanding of the Enron scandal but also demonstrates the potential of graph theory in investigative research in corporate governance and ethics.

Find the code at:

<https://github.com/edargham/enron.git>

Table of Contents

[**Section 1: Enron Dataset** 4](#_Toc154216596)

[1.1 Understanding the Enron Dataset 4](#_Toc154216597)

[1.2 Significance of the Enron Dataset in Graph Theory 4](#_Toc154216598)

[**Section 2: Neo4j** 5](#_Toc154216599)

[**Section 3: Handling the Enron dataset** 5](#_Toc154216600)

[**Section 4: Queries Used** 9](#_Toc154216601)

[**Section 5: Conclusion** 18](#_Toc154216602)

[**Section 6: References** 19](#_Toc154216603)

# **Section 1: Enron Dataset**

Enron Corporation was an American energy, commodities, and services company that became infamous for one of the most notorious corporate fraud cases in history. Founded in 1985 and headquartered in Houston, Texas, Enron was initially involved in energy trading, natural gas, electricity, and communications.

The Enron dataset refers to a collection of emails and financial data acquired from the Enron Corporation, a company that infamously collapsed in 2001 due to widespread corporate fraud. This dataset is highly regarded in research and analytical circles for its comprehensive insights into corporate communication, financial records, and unethical business practices.

## 1.1 Understanding the Enron Dataset

The dataset primarily comprises approximately 500,000 emails exchanged among Enron employees, encompassing a wide spectrum of topics such as business dealings, personal interactions, financial discussions, and more. Additionally, it may include financial statements, balance sheets, and other accounting records, shedding light on Enron's financial status and the fraudulent accounting techniques employed to manipulate financial data.

## 1.2 Significance of the Enron Dataset in Graph Theory

Graph theory involves the study of graphs as mathematical structures to model relationships between entities. The Enron dataset, with its vast network of communication among employees, serves as a rich source for constructing graphs to represent relationships between individuals, departments, or topics based on email exchanges.

# **Section 2: Neo4j**

To integrate the data and have the best analysis we used Neo4j. Neo4j is a popular graph database management system that facilitates the storage and analysis of graph data. Its capabilities make it well-suited for handling interconnected datasets like the Enron emails, enabling graph-based analysis, pattern recognition, and visualization of complex relationships within the dataset. The decision to use the Enron dataset with Neo4j for a graph theory project stems from the dataset's intrinsic characteristics as a large-scale, interconnected communication network. By leveraging Neo4j's graph database functionalities, it becomes feasible to explore email relationships, identify patterns, and uncover insights into communication structures within Enron.

# **Section 3: Handling the Enron dataset**

Since the Enron file we got is a csv file with just the emails, we cannot directly use Neo4j instead we had to clean and handle the data in a specific format in order to save them in the neo4j database. To access and use the needed information from the dataset, we used a python code that read these emails and saves them. First we read the csv file using the following python command:

data = pd.read\_csv(csv\_path, usecols=['message'])

data['id'] = data.index

Which gave the following result:

As seen the above result shows the first and last five rows in the csv file which gave us the message as one column. Therefore we need to extract usedful information from these messages. To do so, we used:

  data[['sender', 'recipient', 'subject', 'cc', 'body']] = data['message'].apply(

        lambda email: pd.Series(parse\_email(email))

    )

    print('Processing Emails. This may take a couple of minutes...')

    data.drop(columns=['message'], inplace=True)

    data['sender'] = data['sender'].apply(sanitize\_email\_address)

    data['recipient'] = data['recipient'].apply(filter\_recipient\_email)

    data['cc'] = data['cc'].apply(filter\_recipient\_email)

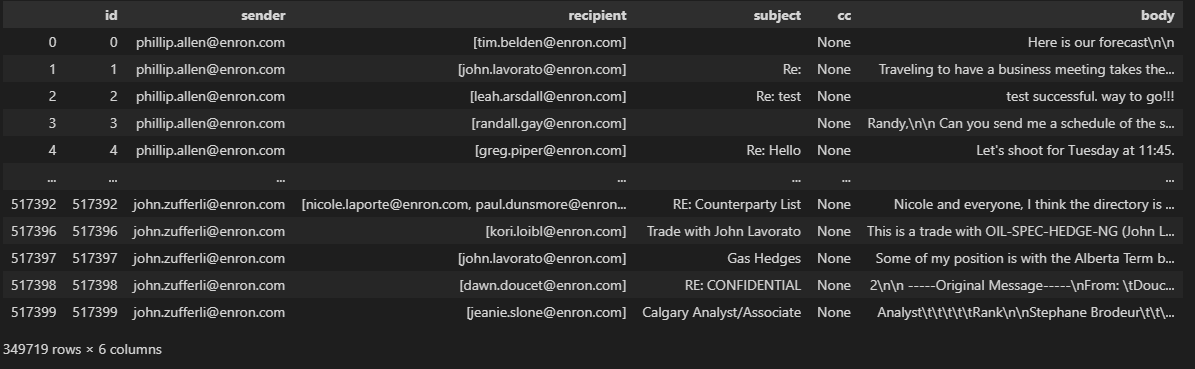
    data = data[data['recipient'] != 'None']

    data = data[data['sender'].str.contains('@enron.com')]

sanitize\_email\_address and filter\_recipient\_email are functions we wrote in order to clean and handle the information. Where “sanitize\_email\_address” takes an email address string (address) as input and aims to sanitize or clean it. The purpose of this function is to extract and return the valid email address from a given string, if it exists. If the input string does not contain a valid email address, it returns the original string. And “filter\_recipient\_email” processes a string of comma-separated email addresses, filters out the ones containing '@enron.com', sanitizes them using sanitize\_email\_address, and returns either the list of filtered email addresses or the string 'None' if there are no matches or if the input is None.

Then we took none null data as to reduce errors in nodes and lastly we made sure the emails we took are only employees by only taking emails that contain “@enron.com”. This is done to the purpose of only using a subset of the data with critical emails for analysis since what is most needed is the emails sent between employees in enron and not outsiders.

After splitting the message into useful information, we got the following data:



Column message was dropped since it was not needed anymore.

After cleaning and handling the data into the needed format. Establishing the connection between our python code and neo4j was the next step. First we created database in neo4j dbms called enron and then using the function

def connect\_to\_n4j():

    db.set\_connection(f'{config.DATABASE\_URL}/{config.DATABASE\_NAME}')

# endfunc

This function intends to set up a connection to a Neo4j database using parameters defined in the config module and using a URL derived from config.DATABASE\_URL and config.DATABASE\_NAME. To be able to establish the connection the following packages were imported “from neomodel import config, db”

config.DATABASE\_URL = app\_config['db\_connection']

config.DATABASE\_NAME = app\_config['db\_name']

 employees = set()

        for sender in data['sender'].to\_list():

            employees.add(sender)

        # endfor

        for recipients in data['recipient'].to\_list():

            for r in recipients:

                employees.add(r)

            # endfor

        # endfor

        for cced in data['cc'].to\_list():

            if cced != 'None':

                for cc in cced:

                    employees.add(cc)

                # endfor

            # endfor

        # endfor

The purpose of this code is to aggregate all unique email addresses from different columns ('sender', 'recipient', 'cc') of the DataFrame data into a set named employees. This set will contain all unique email addresses associated with this dataset,

 for employee in employees:

            ename = employee.split('@')[0]

            emp = Employee.nodes.get\_or\_none(emp\_name=ename)

            if emp is None:

                print(f'Adding Employee {ename} to database.')

                emp = Employee(emp\_name=ename, address=employee).save()

            # endif

            else:

                print(f'Employee {ename} already exists in the database (id: {emp.uid}).')

            # endelse

        #endfor

To add employees to neo4j as nodes, the following code is needed to iterate through the email addresses stored in the employees set, checks if corresponding employees exist in the database based on their names, and adds them to the database if they don't exist, or logs their existence if they are already present.

Similarly for the email message where it contain the email address as node, mid, sender recipients, subject, cc and body as attributes related to the email message.

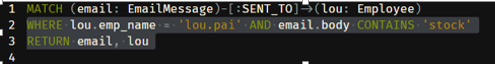
Lastly to set relationships between the employees and email messages we added code that performs intricate operations to construct a comprehensive representation of an email communication network within a Neo4j graph database. Subsequently, it meticulously establishes relationships between these nodes to illustrate the connections in the communication network. The code handles sender, recipient, and CC details by associating 'Employee' nodes with 'EmailMessage' nodes using relationships ('SENT\_FROM', 'SENT\_TO', 'SENT\_CC'). Specifically, it identifies sender 'Employee' nodes and forms 'SENT\_FROM' relationships with corresponding 'EmailMessage' nodes, signifying the emails sent by specific individuals. For recipients and CCed individuals, it recognizes their 'Employee' nodes and creates 'SENT\_TO' and 'SENT\_CC' relationships, respectively, showcasing the recipients and CCed parties of each email. Throughout this process, the code logs the establishment of these connections, providing insights into the network's structure and email interactions. This systematic approach of node creation and relationship establishment enables the representation of email communication patterns, allowing for a detailed analysis of interactions and associations within the communication network stored in the Neo4j graph database.

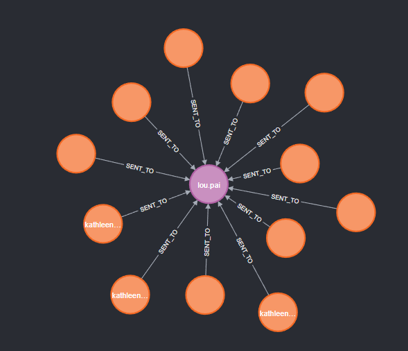
This process took around 8-10 hours to seed the entire database, and took around 1.2 GB of storage space.

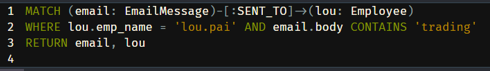
Upon successfully integrating all the email communication data into Neo4j, We shifted and started working on queries to find relationships between data and figure what were the main reasons of Enron’s downfall and fraud. For the code it was a contribution between all team members however, since Neo4j did not work on all computers and some were unable to add the information from python to neo4j, we worked together and sent the python code to one computer, to Elias, in order to run the code and check its work. As for the queries we each came up and wrote several ones and tried them.

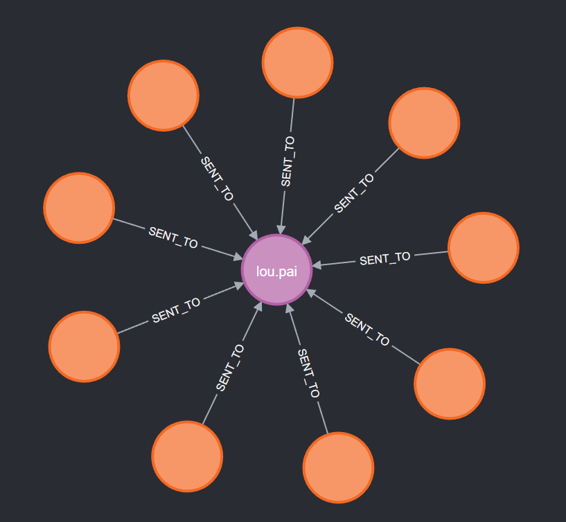
# **Section 4: Queries Used**

As for Queries, we looked for information about Enron’s fraud and found several people related. However, what peaked our interest was an instagram real about an ex-employee in Enron called Lou pai. To investigate him we tried several queries such as:

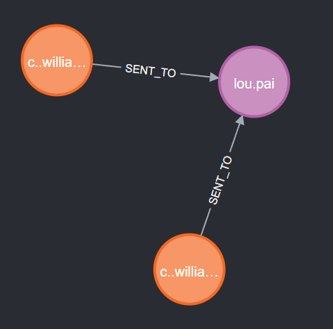


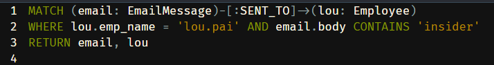


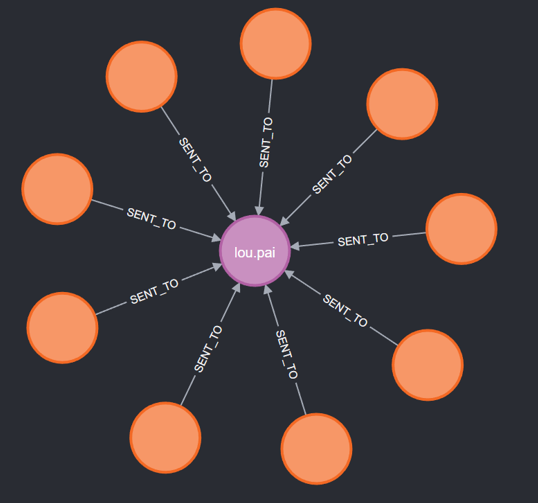












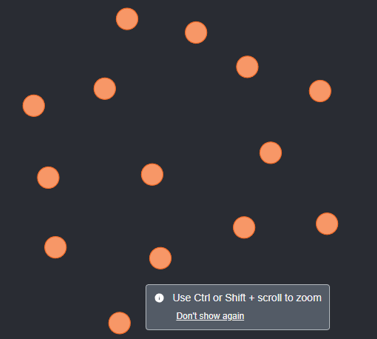
These queries show that Lou pai is related to topics containing insider, stock, trading, notice... Which are one of the main reasons for Enron’s collapse and fraud. Therefore, more investigation were needed and found out that Pai ran Enron's retail energy unit, arriving at the company after stints at ConocoPhillips and DuPont. He left Enron six months before it collapsed in late 2001, pocketing more than $265 million from exercising Enron options and selling stock. Moreover, while further Investingting LOU Pai these emails were found sent by Rex Rogers:

“body: I have been asked to make a brief presentation at next Monday,s Executive   
Committee meeting addressing a new S.E.C. insider trading rule.  Although the   
new rule may increase exposure to liability for insider trading, certain   
provisions of the new rule may actually provide for greater flexibility in   
the timing of your personal trades in Enron Corp. common stock.  Attached is   
a short memo addressing our current Company procedures and policies for   
trading, the new S.E.C. rule, and some suggestions for alternatives that you   
may want to consider concerning your personal trades in Enron Corp. common   
stock.  If anyone wants to discuss the new rule and the trading alternatives   
provided by the new rule before next week,s meeting, please don,t hesitate to   
give me a call at 713-853-3069.  Thank you.

Attachments”

Since insider trading was one of the reasons for the collapse, we looked into further emails containing S.E.C. insider trading rule which resulted in several emails about this topic as seen below



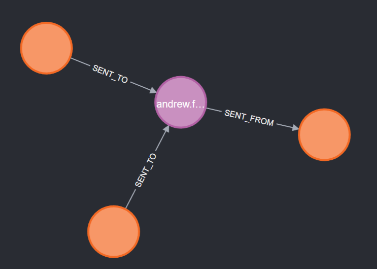


Since Rex Rogers was the sender of these types of emails “S.E.C. Rules”, we looked into him and found out that Rex R. Rogers was a former associate general counsel and vice president of Enron. The Commission filed charges against Mintz and Rogers on March 28, 2007, alleging that they had engaged in a fraudulent scheme to conceal Enron's related-party transactions involving partnerships under the supervision of Andrew Fastow, the company's chief financial officer, as well as the payments Fastow had received as a result of those transactions. Rogers also concealed Enron's related-party transactions including insider stock sales by Kenneth Lay, the company's chairman, as part of the alleged scheme.

Moreover, “In settlement of this action, Mintz and Rogers neither admitted nor denied the allegations of the Commission's complaint. Among other things, the complaint alleged the following: In 1999, Enron sold an interest in a troubled power project in Cuiaba, Brazil to a related party called LJM1, a partnership controlled by Fastow, to deconsolidate the project and recognize related earnings. Under accounting rules, deconsolidation and earnings recognition were inappropriate because Enron did not transfer the risks and rewards of ownership in light of a secret side agreement promising that LJM1 would not lose money on Cuiaba. Satisfying the side agreement, Mintz helped Enron repurchase Cuiaba from LJM1 in 2001. Mintz then delayed signing and closing of the Cuiaba buyback in an effort to avoid reporting related-party transactions in Enron's 2000 Proxy Statement and 2001 Second Quarter Form 10-Q. Moreover, Mintz and Rogers failed to disclose in Enron's 2000 Proxy Statement millions of dollars Fastow received through related-party transactions between LJM and Enron. Rogers further failed to disclose in Enron's 2000 Proxy Statement at least $16 million in insider stock sales by Chairman Kenneth Lay to repay his Enron line of credit during 2000, and aided and abetted Lay's failure to disclose in SEC Form 4 filings an additional $70 million in insider stock sales by Lay during 2001” (Jordan H. Mintz and Rex R. Rogers, U.S. SECURITIES AND EXCHANGE COMMISSION, March 28, 2007).

As seen in the above information, what relates rogers, Fastow, and the ceo at that time Kenneth Lay is the inside trading such as LJM. For that reason, we looked into email relating Fastow and LJM partnerships.





Looking into these email messages, we found the following information about Enron and LJM partnership:

“LJM partnerships and the Enron Wholesale Services (EWS). Here's how the LJM affair led to the EWS investigation:

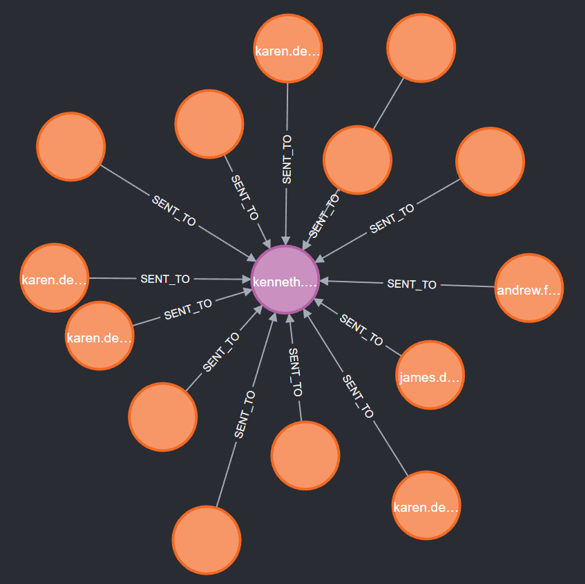
1. **LJM Partnerships**: LJM1 and LJM2 were investment partnerships created by Enron's then-CFO, Andrew Fastow. These partnerships were named after his wife and children's initials. They were designed to buy Enron's poorly performing assets and hedge risky investments, ostensibly to help Enron maintain its credit rating and stock price. However, these partnerships were used to hide Enron's mounting debt and inflate its profits.
2. **Conflict of Interest and Financial Manipulation**: Fastow, as the CFO of Enron and the managing partner of LJM, was in a clear conflict of interest. LJM's transactions with Enron were not conducted at arm's length, allowing Enron to sell underperforming assets to LJM at inflated prices. This deceitful practice helped Enron report financial figures that were much rosier than the reality.
3. **Discovery and Escalation**: As investigators and analysts began to scrutinize Enron’s financial statements and partnerships like LJM, they discovered numerous irregularities. The complexity and opacity of these transactions, along with whistleblowers' testimonies, brought significant attention to Enron’s accounting practices.
4. **Enron Wholesale Services (EWS) Investigation**: The investigation into LJM led to broader scrutiny of Enron’s entire operations, including Enron Wholesale Services. EWS was a core segment of Enron's business, dealing with commodities trading and energy services. Investigators found that not only were the LJM partnerships deceptive, but many of Enron’s divisions, including EWS, were involved in questionable accounting practices. This included "mark-to-market" accounting, where anticipated future profits from energy contracts were booked immediately, often inflating the actual revenue figures.
5. **Collapse of Enron**: The revelations of the scale of fraud within Enron, including the activities involving LJM and the problematic practices in divisions like EWS, led to a loss of investor and market confidence. This culminated in Enron's bankruptcy in December 2001, which was the largest bankruptcy in U.S. history at that time.

The Enron scandal, including the LJM affair and the EWS investigation, became a textbook example of corporate fraud and led to significant changes in corporate governance and accounting regulations, including the enactment of the Sarbanes-Oxley Act in 2002.”

Some information we found that “Andrew S. Fastow, Enron's former Chief Financial Officer. The complaint, filed on October 2, 2002 in U.S. District Court in Houston, alleged that Fastow defrauded Enron's shareholders and enriched himself and others by, among other things, entering into undisclosed side deals, manufacturing earnings for Enron through sham transactions, and inflating the value of Enron's investments.”(Andrew S. Fastow, U.S. SECURITIES AND EXCHANGE COMMISSION)

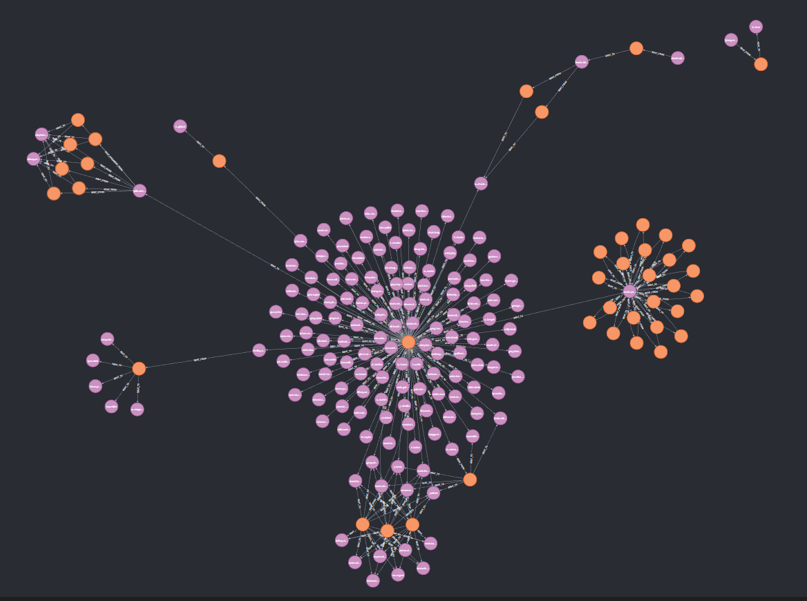
Since Kenneth Lay was the CEO during LJM partnership, we looked for realtion between these two:





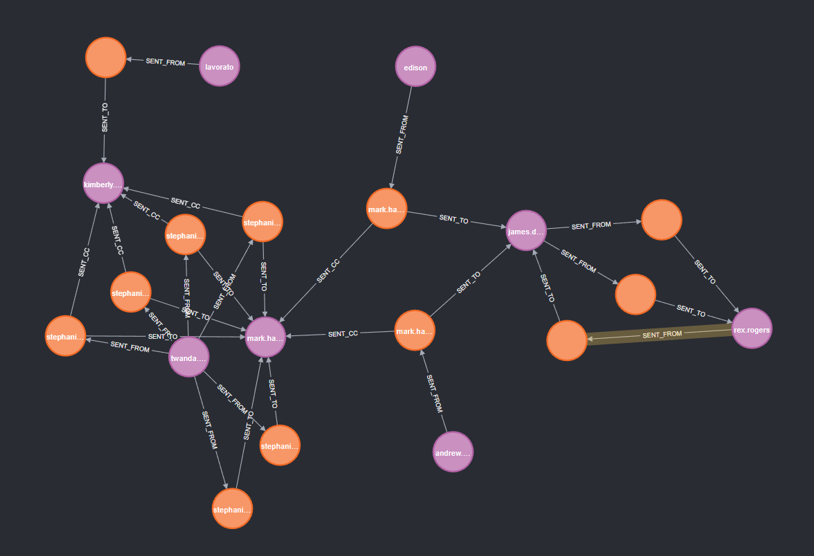
This shows that Kenneth only received emails about LJM and after looking into the messages, most of these emails were articles. Hence we concluded that Kenneth Lay may not have been directly related to this affair. After that we shifted our focus and started investigating relationship between LJM and EWS litigation and found the following:





LJM and EWS litigation are directly connected and as such we looked into emails related to EWS litigation and found the following:





As seen in the above graph, Andrew Fastow and Rex Rogers are also directly related to EWS litigation. Enron Wholesale Services markets and delivers physical commodities and financial risk management services to customers around the world. One of the key strategies used by Enron Wholesale Services involved creating off-balance-sheet entities, such as special purpose entities (SPEs), to conceal debt and losses. These entities were used to manipulate financial figures, making the company appear more profitable and financially stable than it actually was.

# **Section 5: Conclusion**

In conclusion, leveraging a graph database engine for analyzing the Enron dataset has proven instrumental in uncovering valuable insights in the data. The intricate relationships and dependencies among individuals, entities, and communications within the Enron dataset are effectively captured and navigated through the graph database. This approach has enabled a more nuanced understanding of the complex network of interactions, identifying patterns, anomalies, and trends that might be challenging to discern with traditional relational databases. The inherent flexibility of graph databases facilitates the exploration of interconnected data, offering a dynamic and efficient means to extract meaningful information. In the context of the Enron dataset, the graph database's ability to model and traverse relationships has provided a comprehensive view of the organizational dynamics and communication flows, thereby enhancing the depth and accuracy of insights gained. The utilization of graph databases is increasingly pivotal in unraveling complex datasets, making them an invaluable tool for extracting actionable intelligence and improving decision-making processes in diverse domains.

After analyzing and connecting the information we got from the above queries, we discovered that some of the reasons for Enron’s fraud and collapse where related to insider trading mainly by top executives Lou Pai, LJM affair orchestrated by Andrew Fastow which was the main issue that led to the company-wide EWS litigation.

# **Section 6: References**

*SEC.gov | Jordan H. Mintz and Rex R. Rogers*. (2009, January 26). <https://www.sec.gov/litigation/litreleases/lr-20866>

*SEC.gov | Lou L. Pai*. (2008, July 29). <https://www.sec.gov/litigation/litreleases/lr-20658>

*SEC.gov | Andrew S. Fastow*. (2004, January 14). <https://www.sec.gov/litigation/litreleases/lr-18543>

*Enron Wholesale Services*. (n.d.). <https://enroncorp.com/ews/#:~:text=Enron%20Wholesale%20Services&text=Enron%20Wholesale%20Services%20markets%20and,which%20few%20others%20can%20match>.

*Cypher Cheat Sheet - Neo4J Documentation Cheat Sheet*. (n.d.). Neo4j Graph Data Platform. <https://neo4j.com/docs/cypher-cheat-sheet/5/auradb-enterprise/>

*Getting started — neomodel 5.2.0 documentation*. (n.d.). <https://neomodel.readthedocs.io/en/latest/getting_started.html>

*The Enron email Dataset*. (2016, June 16). Kaggle. <https://www.kaggle.com/datasets/wcukierski/enron-email-dataset>

Hayes, A. (2023, March 28). *What was Enron? what happened and who was responsible*. Investopedia. <https://www.investopedia.com/terms/e/enron.asp>

What is a graph database? - Getting Started. (n.d.). Neo4j Graph Data Platform. <https://neo4j.com/docs/getting-started/get-started-with-neo4j/graph-database/#:~:text=A%20Neo4j%20graph%20database%20stores,thinking%20about%20and%20using%20it>.

Walia, M. S. (2023, October 8). *Neo4J: features, how to use, challenges & more*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2022/01/a-comprehensive-guide-on-neo4j-graph-database/>