# Summary

For this assignment, I am tasked to find fraudulent returns within a company, and by association, violations in company policy regarding returns. I make use of a SQL database that includes information about 25 employees as well as all the returns they I did manage to find both instances of said activities. I found three total employees handling returns in a way that goes against company policy and/or in a fraudulent manner.

# Objectives

* Retrieve hash number from database file for comparison.
* Query the data to search for missing returns information.
* Query the data to search for employees creating their own returns.
* Query the data to search for employees not following policy regarding returns.
* Query the data to search for matching credit card numbers.

# Chain of Custody

|  |  |
| --- | --- |
| Original Hash Number | 36821523F51EE71F48D50813CB9F5B979C9111745133C95F2348830C9BD56B71 |
| Retrieved Hash Number | 36821523F51EE71F48D50813CB9F5B979C9111745133C95F2348830C9BD56B71 |
| Date Created | 2/20/24 at 6:06:49 PM |
| Date Retrieved | 2/20/24 at 6:06:49 PM |

# Methods and Tools Used

Date of File:

For this step, I simply inspected the properties of the file to view the date it was created as well as when I retrieved it.

Google Colab Code (for hash number):

from google.colab import files

import hashlib

def calculate\_hash(file\_content):

    # Calculate the hash using the SHA-256 algorithm

    hash\_value = hashlib.sha256(file\_content).hexdigest()

    return hash\_value

# Upload the file

uploaded = files.upload()

# Get the file name and content

file\_name = next(iter(uploaded))

file\_content = uploaded[file\_name]

# Calculate the hash

hash\_value = calculate\_hash(file\_content)

print("SHA-256 Hash:", hash\_value

Google Colab File:

<https://colab.research.google.com/drive/1VMsitprTyK7L31cAjV73D6nqc-eiwveu>

SQL Queries:

SELECT \* FROM employee;

SELECT \* FROM returns;

SELECT \*

FROM returns

WHERE

(ReturnDate IS NULL OR ReturnDate = '') OR

(ReturnPrice IS NULL OR ReturnPrice = '') OR

(CustomerName IS NULL OR CustomerName = '') OR

(Street IS NULL OR Street = '') OR

(City IS NULL OR City = '') OR

(State IS NULL OR State = '') OR

(CustomerPhone IS NULL OR CustomerPhone = '')

ORDER BY ReturnPrice DESC;

SELECT \*

FROM employee e

INNER JOIN returns r ON e.Name = r.CustomerName;

SELECT \*

FROM returns

WHERE IsReceiptPresent = 'False'

AND GiftCardNumber IS NULL

AND EmployeeID != 7857

ORDER BY ReturnPrice DESC;

SELECT CreditCardNum, COUNT(\*) AS NumEntries

FROM returns

GROUP BY CreditCardNum;

SELECT \* FROM returns

WHERE CreditCardNum = 2703144240071235;

SELECT \* FROM returns

WHERE CreditCardNum = 40077653923591;

SELECT \* FROM returns

WHERE EmployeeId = 7857;

SELECT \* FROM returns

WHERE EmployeeId = 9663;

SELECT \* FROM returns

WHERE EmployeeId = 3277;

# Relevant Findings

* One of the first things I checked for was an employee’s name being the same as a customer on a return. I did find one instance of this, the person being Kathryn Anderson. The only problem is that they reside in entirely different places and the employee ID does not match up.
* I checked for returns where there was no receipt and no gift card number. There were 85 rows of this, 68 of which were with 7857 (Mark Moore) as the associated employee ID. I excluded him from this query for reasons I will address in the next point. This left me with 17 entries made up of two employees with IDs of 3277 (John Mason) and 9663 (Anthony West Jr.). For these two employees, this activity represents violations of company policy more than anything in my opinion. 13 of the 17 were from John, he has a total of 56 returns as well. Anthony on the other hand has 43 in total. This information could make a difference in the necessary punishment between the two.
* One major finding was one credit card number having multiple entries. There were two credit card numbers like this, 2703144240071235 and 40077653923591. I checked to make sure these were not just one person making more than one return, and this was not the case. Not only were all the customer identities unique, but the associated employee ID was the same for all 68 returns. Additionally, these 68 returns are out of 167 returns involving this employee, just under half. This points to fraudulent activity for the employee named Mark Moore with the ID 7857.