Financial Fraud Analysis: Database Design and Analytics

BSAN 726

December 15, 2024

Group 3

Sakkyra King

Mahsa Karbalaei

Haley Karolevitz

Instructor

Professor Karthik Srinivasan

Introduction

Financial fraud is an ever-growing concern that significantly impacts businesses and individuals alike, resulting in billions of dollars in losses annually. With the increasing reliance on digital platforms and online transactions, fraudulent activities have become more sophisticated, making it a top priority for financial institutions to detect and mitigate such risks effectively. Beyond monetary losses, fraud undermines customer trust, tarnishes reputations, and disrupts business operations, highlighting the importance of proactive fraud detection strategies.

Detecting fraud requires a data-driven approach, as financial transactions generate vast amounts of information that can reveal suspicious patterns and behaviors. By utilizing advanced database structures and analytical techniques, businesses can uncover actionable insights to identify and mitigate fraudulent activities. These tools not only improve operational efficiency but also empower organizations to implement real-time solutions, reducing risk and enhancing overall security.

In this project, we focus on designing a fraud detection framework that leverages structured datasets and targeted analytics. Through strategic database modeling and the application of SQL queries, we aim to provide a clear, scalable solution for identifying trends, high-risk entities, and transaction characteristics that contribute to fraud. This project underscores the critical role of data analytics in safeguarding financial systems while addressing the challenges and opportunities of fraud prevention in today's dynamic business landscape.

Problem Statement and Questions

We aim to analyze financial transaction data to identify potentially fraudulent activities. By determining if certain transactions have relations to fraud and recognizing common patterns, we want to enhance the business' capabilities to proactively detect and

prevent fraudulent transactions in the future. This report outlines the key problems and questions, proposed solutions, initial challenges, and an overview of the data that will be utilized for this fraud detection effort.

The primary goal of this project is to ascertain whether financial transactions processed by the company have any relationships to fraudulent activities, and to identify specific patterns that can help understand the characteristics most frequently associated with confirmed fraudulent transactions. This knowledge will be leveraged to improve fraud detection models and processes moving forward. To guide the analysis, several key questions have been raised:

1. Which customer segments or profiles are most often linked to fraudulent transactions?
2. What are the peak days, times and seasons when fraudulent transactions tend to occur?
3. Which merchants are frequently involved in suspicious transactions?
4. How do internally generated anomaly scores correlate with actual confirmed fraud occurrences?

Answering these questions will require a deep dive into transaction records, customer information, merchant data and historical fraud patterns. Connecting data across these domains will be critical to uncovering meaningful insights.

With a clear understanding of the problem, a pragmatic data and analysis approach, and eyes wide open to the challenges ahead, our team is well positioned to unlock valuable insights from its financial data to enhance fraud detection capabilities. While it won't be an easy undertaking, the operational and financial benefits of fewer losses, reduced false positives, and improved customer experience make it a worthwhile investment. The multi-layered anomaly detection analysis approach powered by a robust, comprehensive and linked data foundation will enable the company to identify a wider range of suspicious activities more accurately and efficiently than ever before. As fraudsters grow increasingly sophisticated, this data-driven, adaptive fraud detection

capability will ensure the company stays a step ahead to protect its assets and its customers.

Inital Challenges

Here are some initial challenges we had when working with data set. When making sense of the data, it was divided into 10 files that had to be combined. It's important to ensure the data remains accurate, reliable, and consistent during this process. All pieces must be connected to uncover patterns of fraud. SQL queries on the raw data will provide results that need careful interpretation. The team must extract meaningful insights without rushing to conclusions. Multiple queries will likely be needed to build on each other and better understand fraud patterns. When finding the right balance, flagging too many legitimate transactions as fraud can cause customer dissatisfaction and inefficiency, while missing actual fraud leads to financial losses. The models need to strike the right balance by minimizing losses and keeping investigation costs reasonable. These tradeoffs must be carefully measured and presented to support effective business decisions.

Proposed Solutions

To enable this fraud analysis initiative, a multi-pronged data management and analysis approach is proposed. In terms of data management, Microsoft Excel will be used for initial data exploration and preparation. SQL will serve as the foundation for a larger-scale relational database to efficiently store and query the data. Tables will be linked using foreign keys to ensure referential integrity and seamless analysis across data entities. The data will be organized into specific tables focused on fraud indicators, which include metrics and attributes historically associated with fraud, suspicious activity that captures specific transactions or series of events flagged as potential fraud, customer profiles containing demographic and behavioral information, account activity that details financial

accounts and their usage, and merchant information that includes profiles and transaction histories of merchants accepting payments. An Online Transaction Processing (OLTP) architecture using a SQL database will enable the management of data on customers, merchants, accounts, and transactions. It will also allow real-time processing for time-sensitive fraud detection and efficient querying to support the required analysis.

For the analytical approach, the fraud detection analysis will focus on a few key optimization goals. Reducing false positives, where legitimate transactions are incorrectly flagged as fraud, is crucial to minimize customer friction and unnecessary investigation effort. At the same time, minimizing false negatives, which occur when actual fraudulent transactions are missed, will help limit financial losses. Fine-tuning anomaly thresholds is necessary to achieve an optimal balance between casting a wide enough net and maintaining precision. Additionally, incorporating the costs associated with fraud will enable business-driven decisions on risk management. This optimization approach will be applied across the data to build and refine fraud detection models.

Kaggle Fraud Detection Database

The dataset we utilized for this project was the Fraud Detection Database from Kaggle, which consisted of five primary folders, each containing two files. These folders included Fraudulent Patterns, Customer Profiles, Merchant Information, Transaction Amounts, and Transaction Data. Within these folders, key files such as account activity, customer and merchant data, transaction records, and anomaly scores provided valuable information for our analysis. We selected this dataset because it reflects the essential information that banks and organizations monitor for fraud detection. This made it a suitable foundation for our project.

Before constructing the ER model and data warehouse, we cleaned the dataset to align with our goals. First, we removed the Fraudulent Patterns folder. Since our objective was to detect potential fraud using transaction data, having a folder that pre-identified

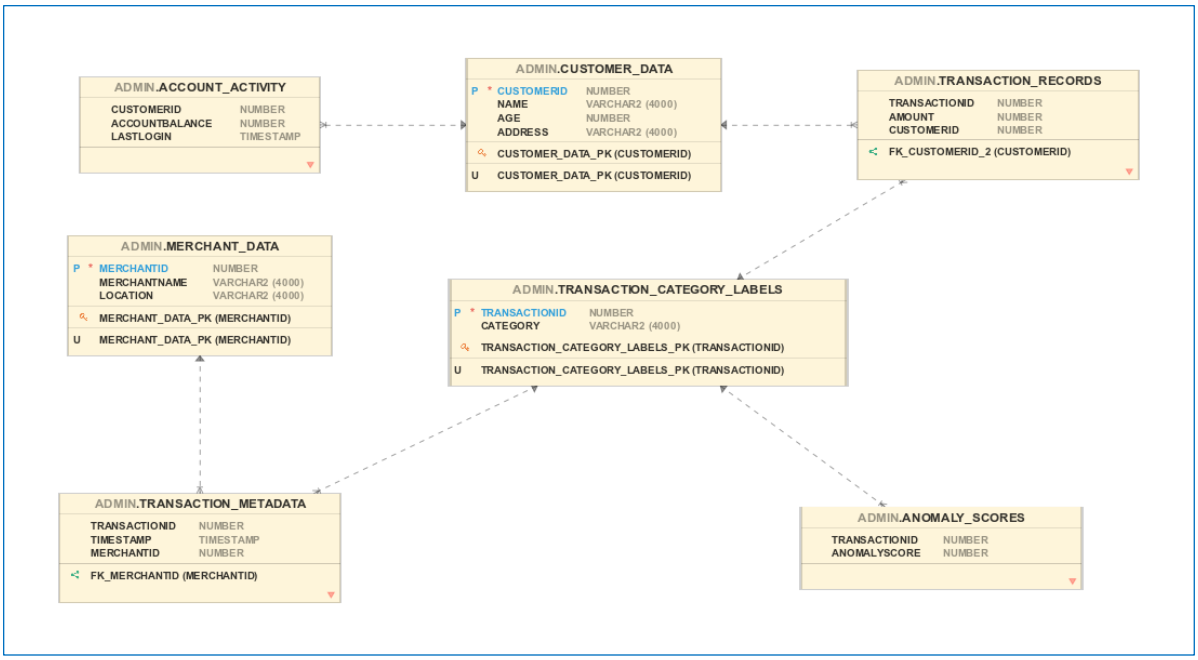
fraud would undermine the analytical process. Instead, we aimed to draw conclusions from other data points, such as transaction records and anomaly scores, without relying on predefined outcomes. This adjustment ensured the dataset fits the scope of our business problem.

Additionally, we removed the Transaction Amounts file, as this data was already included within the Transaction Records file. Retaining both would have introduced redundancy, so removing the duplicate file streamlined the dataset while preserving all necessary information for analysis.

Conceptual Model: ER Model

When creating the ER model for the fraud detection dataset, we identified the entities based on the folders they were assigned to. These entities represent the key pieces of information required to analyze the data effectively. The main entities we identified include Customer Data, Account Activity, Merchant Data, Anomaly Scores, Transaction Records, Metadata, and Transaction Category. The relationships between these entities are as follows: Customer and Transaction Records, Merchant and Transaction Metadata, Customer and Account Activity, and Transaction Records along with Transaction Metadata and Anomaly Scores linking to Transaction Category.

Customer and Transaction Records will have a one-to-many relationship, as customers can have multiple transactions, but each transaction is associated with only one customer. This same type of relationship applies to Merchant and Transaction Records, Customer and Account Activity, and Account Activity and Transaction Records. By defining these entities and their relationships, we established a clear and organized structure for the data, ensuring that the ER model captures the necessary connections to support fraud detection analysis.



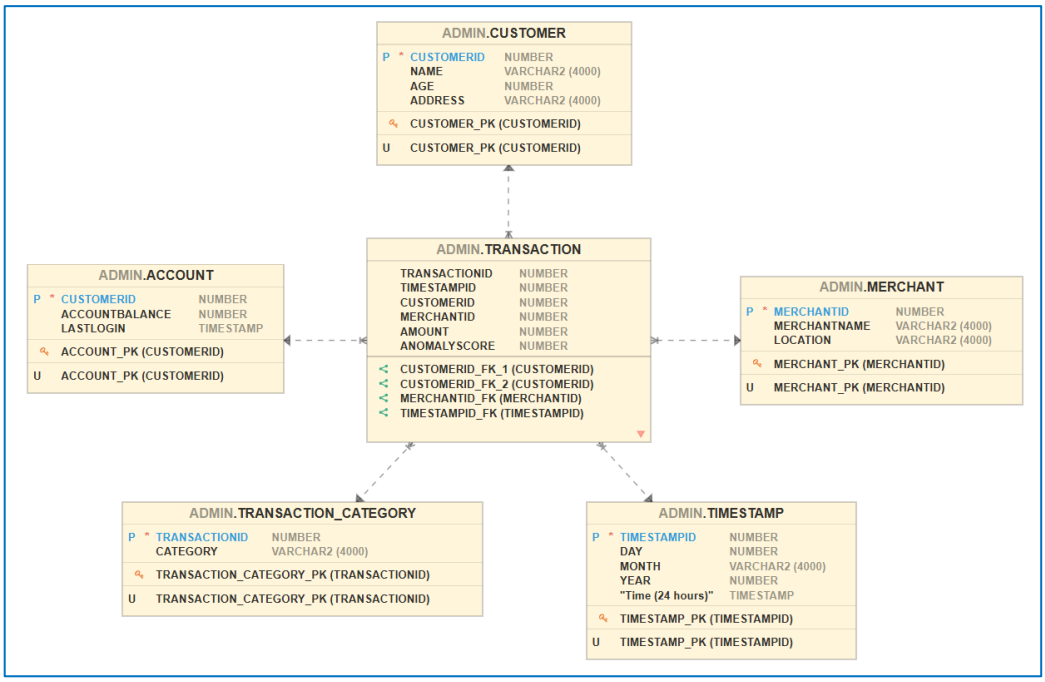
Conceptual Model: Data Warehouse

To optimize our fraud detection analysis, we structured the data warehouse using a Star Schema, a design that streamlines data organization and enhances query performance. In this design, the Fact table is Transactions, which includes two key facts—Amount and AnomalyScore—as well as four dimension foreign keys: merchantID, transactionID, timestampID, and customerID. The Amount fact captures the monetary value of a transaction, while the AnomalyScore indicates the likelihood, expressed as a percentage, that the transaction could be fraudulent.

Surrounding the Fact table are five Dimension tables: Customer, Account, Merchant, Transaction Category, and Timestamp. The Customer table includes customerID along with attributes such as name, age, and address. The Merchant table contains merchantID, merchant name, and location. The Account table connects customerID with account details, including account balance and the last login timestamp (indicating the most recent account access). The Transaction Category table includes transactionID and

the category of the purchased item. Lastly, the Timestamp table represents the grain, breaking down the transaction time into day, month, year, and exact time.

To develop this model, we aligned it with our business processes and questions, ensuring the structure supports targeted analysis. This data warehouse design provides the foundation to answer the given question listed earlier, enabling efficient analysis of trends, patterns, and risks related to fraudulent activities.



Proof of Concepts: Testing Business Queries

We implemented a series of SQL queries to uncover meaningful fraud detection insights that answer our business questions. Our queries were designed to highlight patterns in fraudulent activity. We focused on identifying high-risk merchants and customers, peak fraud hours, and how transaction characteristics influence risk. The relational database structure provided a strong foundation for linking and analyzing our data through SQL operations.

```sql
--- Query #1
SELECT m.MerchantID,
       m.MerchantName,
       COUNT(t.TransactionID) AS HighRiskTransactions,
       AVG(t.AnomalyScore) AS AvgAnomalyScore
FROM ADMIN.MERCHANT m
JOIN ADMIN.TRANSACTION t ON m.MerchantID = t.MerchantID
WHERE t.AnomalyScore > 0.8
GROUP BY m.MerchantID, m.MerchantName
ORDER BY HighRiskTransactions DESC;
```

	MERCHANTID	MERCHANTNAME	HIGHRISKTRANSACTIONS	AVGANOMALYSCORE
1	2844	Merchant 2844	2	0.8936203975
2	2901	Merchant 2901	2	0.839693863
3	2530	Merchant 2530	2	0.9353339795
4	2229	Merchant 2229	2	0.950662322
5	2408	Merchant 2408	2	0.8704318305
6	2154	Merchant 2154	2	0.921913792
7	2587	Merchant 2587	1	0.880078252
8	2590	Merchant 2590	1	0.832788084
9	2625	Merchant 2625	1	0.916374892
10	2629	Merchant 2629	1	0.9603636
11	2638	Merchant 2638	1	0.918156008
12	2643	Merchant 2643	1	0.882698778
13	2648	Merchant 2648	1	0.979642139
14	2652	Merchant 2652	1	0.957631359
15	2671	Merchant 2671	1	0.966645464

Our first query focused on highlighting merchants that are most frequently associated with fraud (high-risk transactions). To do this we joined the Transaction and Merchant tables, isolating transactions with anomaly scores greater than 0.8. We calculated both the number of flagged transactions and their average anomaly scores by grouping the data by merchant ID. Our results highlight the specific merchants that consistently appeared in high-risk scenarios, who should be monitored more closely.

```sql
--- Query #2
SELECT TO_CHAR(t.Timestamp, 'HH24') AS Hour,
       COUNT(t.TransactionID) AS HighRiskTransactions
FROM ADMIN.TRANSACTION t
WHERE t.AnomalyScore > 0.8
GROUP BY TO_CHAR(t.Timestamp, 'HH24')
ORDER BY HighRiskTransactions DESC;
```

	HOUR	HIGHRISKTRANSACTIONS
1	10 10 AM	7
2	14 2 PM	7
3	09 9 AM	6
4	13 1 PM	6
5	23	5
6	18	5
7	02	5
8	15	5
9	05	5
10	03	5

Our second query focused on identifying the peak hours of fraudulent transactions. To do this we extracted the hour from each transaction's timestamp and grouped transactions with anomaly scores above 0.8. Our query identified peak fraud times that align with typical business hours including 10 AM, 1 PM, and 2 PM. There are also high fraud

risks during late night hours like 2 AM, 3 AM, and 5 AM. Increased monitoring during these times could help improve fraud detection efforts and reduce risk.

```sql
--- Q3
SELECT
    CASE
        WHEN t.Amount < 15 THEN 'Low'
        WHEN t.Amount BETWEEN 15 AND 40 THEN 'Medium'
        ELSE 'High'
    END AS AmountRange,
    COUNT(t.TransactionID) AS HighRiskTransactions,
    AVG(t.AnomalyScore) AS AvgAnomalyScore
FROM ADMIN.TRANSACTION t
WHERE t.AnomalyScore > 0.8
GROUP BY
    CASE
        WHEN t.Amount < 15 THEN 'Low'
        WHEN t.Amount BETWEEN 15 AND 40 THEN 'Medium'
        ELSE 'High'
    END
ORDER BY HighRiskTransactions DESC;
```

AMOUNTRANGE	HIGHRISKTRANSACTIONS	AVGANOMALYSCORE
High	74	0.9046561591216217
Medium	19	0.8897718681052632
Low	10	0.8832236867

Our third query focused on evaluating how transaction amounts relate to fraud risk. We categorized transactions into value ranges “Low”, “Medium”, and “High” with all anomaly scores above 0.8. Our query revealed that the most fraudulent transactions occurred in the “High” value range, where amounts are greater than $40. This connection between transaction value and fraud likelihood shows the importance of real-time monitoring for higher value transactions. Implementing real-time alerts for high value transactions that are more prone to fraud could help reduce more significant losses.

```sql
--- Query #4
SELECT tc.Category,
       COUNT(t.TransactionID) AS HighRiskTransactions,
       AVG(t.AnomalyScore) AS AvgAnomalyScore
FROM ADMIN.TRANSACTION_CATEGORY tc
JOIN ADMIN.TRANSACTION t ON tc.TransactionID = t.TransactionID
WHERE t.AnomalyScore > 0.8
GROUP BY tc.Category
ORDER BY HighRiskTransactions DESC;
```

CATEGORY	HIGHRISKTRANSACTIONS	AVGANOMALYSCORE
Online	31	0.8927469761290323
Travel	20	0.90205227725
Retail	19	0.8964343032631579
Other	17	0.8973679687647059
Food	16	0.91742181875

Our fourth query focused on identifying the high-risk transaction categories. We joined the Transaction and Transaction_Category tables and grouped the flagged transactions by their type. Our query showed that categories Online, Travel, and Retail have

higher risks of fraud. This tells us that these specific types of transactions require stricter verification processes and monitoring.

```sql
--- Query #5
SELECT c.CustomerID,
       c.Name AS CustomerName,
       COUNT(t.TransactionID) AS HighRiskTransactions,
       AVG(t.AnomalyScore) AS AvgAnomalyScore
FROM ADMIN.CUSTOMER c
JOIN ADMIN.TRANSACTION t ON c.CustomerID = t.CustomerID
WHERE t.AnomalyScore > 0.8
GROUP BY c.CustomerID, c.Name
ORDER BY HighRiskTransactions DESC;
```

	CUSTOMERID	NAME	HIGHRISKTRANSACTIONS	AVGANOMALYSCORE
1	1796	Customer 1796	2	0.867977159
2	1800	Customer 1800	2	0.834769655
3	1506	Customer 1506	2	0.8763129625
4	1495	Customer 1495	2	0.872095465
5	1244	Customer 1244	2	0.847303652
6	1549	Customer 1549	1	0.832788084
7	1551	Customer 1551	1	0.943482558
8	1555	Customer 1555	1	0.925980753
9	1558	Customer 1558	1	0.991626339
10	1565	Customer 1565	1	0.838278033
11	1575	Customer 1575	1	0.923014689
12	1577	Customer 1577	1	0.997038134
13	1579	Customer 1579	1	0.950089781
14	1587	Customer 1587	1	0.91088818
15	1591	Customer 1591	1	0.966645464

Our last query focused on identifying high-risk customers. We joined the Transaction and Customer tables to find customers with multiple high-risk transactions. Our results highlight the specific customers that consistently appeared in high-risk scenarios. These customers should be flagged for further investigation to further reduce fraud.

Our queries show how SQL can be used to produce actionable results. By using techniques like joins, filtering, and grouping, we were able to identify patterns in our data that could help stakeholders target fraud more effectively. This process is foundational for fraud detection and could be improved by creating more advanced queries or integrating additional analytical tools.

Alternative Solutions

While SQL provided an accessible and structured approach to analyzing fraud patterns, integrating business intelligence (BI) would significantly enhance insights and usability. Tools like Power BI and Tableau allow you to create interactive dashboards that show data trends dynamically and visually. BI dashboards would allow stakeholders to drill down into specific areas, like transaction categories or time periods, to better understand

the data and its patterns. BI tools foster interactivity, unlike SQL and its static outputs. This makes it easier to identify and share actionable insights quickly. Implementing BI would require an increase in time and effort.

Another approach that we could have taken is using regression models for predictive fraud detection. Logistic regression is suited for binary classification problems like identifying fraudulent transactions. By training a model on labeled data, with features like transaction amount, anomaly score, and merchant & customer details, logistic regression could assign probabilities to each transaction and predict its likelihood of being fraudulent. Linear regression could've also been utilized to identify relationships between transaction characteristics and anomaly scores. This would help quantify how certain variables influence fraud risk. While the regression models offer significant advantages, they require preprocessing and feature selection to ensure the data is clean and well-structured for analysis which can lower efficiency.

A more advanced approach that we could have taken is machine learning algorithms like decision trees or random forest models. These models could enhance fraud detection by identifying non-linear relationships and interactions between the data's features. Machine learning builds upon the strengths of regression and captures even more complex patterns. Implementing these models would require an increased technical expertise and upgraded computational resources. Machine learning would be more suitable for businesses with much larger scale operations. For our project, the focus on SQL and BI aligned with the scope and scale of our dataset, but as an effective next step to deepen our analysis and improve accuracy, we could implement the regression and machine learning models.

Implementation Challenges

One challenge we faced was creating a database structure that was efficient and easy to use. We chose to use the star schema, which worked well for our straightforward queries and helped to simplify our data integration. In a real-world business setting, where there would be a lot more data, businesses might have considered using a snowflake

schema. While this schema normalizes data and saves storage, it also slows down queries and increases management efforts. Given the size and scope of our dataset, we concluded that the star schema had the best balance between simplicity and functionality.

Another challenge we faced was SQL's limitation in presenting our findings in an interactive manner. SQL helped us identify fraud trends effectively, but its results were static. This means that stakeholders would have to interpret outputs with no dynamic exploration or updating. To address this challenge, we incorporated Business Intelligence to create a dashboard that visualized key fraud patterns. This improved usability but required extra effort and data integration. Other tools like Tableau or Qlik have powerful features that are good for large-scale implementation. These tools come with high costs and technical requirements. Power BI was a good middle ground for our project, as it balanced accessibility and functionality without being too complex.

One of the bigger limitations of our project was the scope and robustness of our data. While our dataset was sufficient for building queries and analyzing trends, it lacked the depth and variability that are present in real-world datasets. One example of this was that our data was limited to transactions from only two days of each month. This restricted our ability to observe broader trends or seasonal patterns. Additionally, we found certain indicators, like anomaly scores, were pre-calculated and did not always align with true fraudulent transactions. This limitation reduces the reliability and generalizability of our results.

Key Takeaways & Conclusions

This project showed how structured data and SQL queries can be used to identify fraudulent patterns and gain actionable insights. By using a star schema database design, we organized our data to facilitate queries and analysis. Key insights like identifying high-risk merchants and customers, detecting fraud-prone time periods, and highlighting specific transaction categories helped us find patterns that businesses can utilize to improve their fraud prevention strategies. The additional integration of Power BI brought our analysis to life, making it easier to interpret the trends and communicate our findings to

stakeholders. These tools and methods provided a practical, cost-effective solution for identifying and addressing fraud within the scope of our dataset.

Our analysis also highlighted some key limitations and areas for improvement. The constraints of our dataset, like its limited time range and predefined indicators, restricted our analysis. While SQL and Power BI worked well for static queries and dashboards, scaling these tools for larger datasets and real-time analytics would require more advanced systems. Despite these challenges, this project showed the importance of carefully selecting the appropriate tools and techniques based on the scope of the data and business needs. With additional resources, business could expand on the foundations of our analysis to incorporate predictive models or more robust data sources to enhance their fraud detections and overall decision-making.