

**Debre Berhan University**

**College of Computing**

**Department of Software Engineering**

**Big Data and Business Intelligence**

**Individual Assignment**

by

Hassen Muhammed DBUR/0280 /13

**Submitted to**: Mr. Derbew Felasman(MSc)

Date: Feb 2025

Table of Contents

[Documentation for E-Commerce Fraud Detection Data Processing 3](#_Toc190000272)

[Overview 3](#_Toc190000273)

[Requirements 3](#_Toc190000274)

[Data Description 3](#_Toc190000275)

[Program Flow 4](#_Toc190000276)

[1. Load the Dataset 4](#_Toc190000277)

[2. Data Exploration 4](#_Toc190000278)

[3. Data Cleaning 4](#_Toc190000279)

[4. Database Connection & Table Creation 5](#_Toc190000280)

[5. Load Data into PostgreSQL 5](#_Toc190000281)

[Code Implementation 5](#_Toc190000282)

[Output 7](#_Toc190000283)

[6. Data Visualization and Insights: Transaction and Customer Analysis 11](#_Toc190000284)

[Description of the Bar Chart: 12](#_Toc190000285)

[Description of the Pie Chart: 13](#_Toc190000286)

[Description of the Chart: 14](#_Toc190000287)

[Description of the Line Chart: 15](#_Toc190000288)

[Description of the Map Chart: 17](#_Toc190000289)

[Description of the Column Chart: 19](#_Toc190000290)

[Description of the Donut Chart: 20](#_Toc190000291)

[Conclusion 21](#_Toc190000292)

# Documentation for E-Commerce Fraud Detection Data Processing

# Overview

This program processes a dataset containing e-commerce transactions, cleans the data, and loads it into a PostgreSQL database for further analysis. The dataset includes details such as transaction amounts, customer locations, payment methods, and fraud indicators.

# Requirements

* Python 3.x
* Pandas
* SQLAlchemy
* psycopg2
* PostgreSQL

# Data Description

The dataset consists of 16 columns:

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Data Type |
| transaction\_id | Unique identifier for each transaction | UUID |
| customer\_id | Unique identifier for each customer | UUID |
| transaction\_amount | Amount spent in the transaction | NUMERIC |
| transaction\_date | Date and time of the transaction | TIMESTAMP |
| payment\_method | Method used for payment (e.g., credit card) | VARCHAR(50) |
| product\_category | Category of the purchased product | VARCHAR(100) |
| quantity | Number of items in the transaction | INT |
| customer\_age | Age of the customer | INT |
| customer\_location | Geographical location of the customer | VARCHAR(100) |
| device\_used | Device type used for the transaction | VARCHAR(50) |
| ip\_address | IP address used during the transaction | VARCHAR(50) |
| shipping\_address | Address to which the order was shipped | TEXT |
| billing\_address | Billing address of the customer | TEXT |
| is\_fraudulent | Indicator (0 or 1) of fraudulent transactions | BOOLEAN |
| account\_age\_days | Age of the customer account in days | INT |
| transaction\_hour | Hour of the day the transaction occurred | INT |

# Program Flow

## Load the Dataset

* + Reads data from Fraudulent\_E-Commerce\_Transaction\_Data.csv using Pandas.

## Data Exploration

* + Displays the first five rows of raw data.
  + Prints data types and missing values.

## Data Cleaning

* + Removes duplicate rows.
  + Drops rows with missing values.
  + Converts Transaction Date to a timestamp.
  + Formats column names to be lowercase and PostgreSQL-compatible.

## Database Connection & Table Creation

* + Connects to PostgreSQL using SQLAlchemy.
  + Creates the transactions table if it does not exist.

## Load Data into PostgreSQL

* + Writes cleaned data into the PostgreSQL database.

## Code Implementation

import pandas as pd

import psycopg2

from sqlalchemy import create\_engine, text

# Database connection details

DB\_USER = "smilex"

DB\_PASSWORD = "smilex"

DB\_HOST = "localhost"

DB\_PORT = "5432"

DB\_NAME = "ecommerce\_db"

TABLE\_NAME = "transactions"

# Load dataset

file\_path = "Fraudulent\_E-Commerce\_Transaction\_Data.csv"

df = pd.read\_csv(file\_path)

# Display raw data info

def log\_data\_info(df, stage):

    print(f"\n{stage} Data Snapshot:")

    print(df.head())

    print(df.describe(include='all'))  # Better statistical summary

    print(f"Missing values per column:\n{df.isnull().sum()}")

log\_data\_info(df, "Raw")

# Data Cleaning

df.drop\_duplicates(inplace=True)

df.dropna(inplace=True)

df['transaction\_date'] = pd.to\_datetime(df['transaction\_date'])

df.columns = [col.lower().replace(" ", "\_") for col in df.columns]

# Convert data types

df['is\_fraudulent'] = df['is\_fraudulent'].astype(bool)

df['transaction\_hour'] = df['transaction\_hour'].astype(int)

df['account\_age\_days'] = pd.to\_numeric(df['account\_age\_days'], errors='coerce')

df['transaction\_amount'] = pd.to\_numeric(df['transaction\_amount'], errors='coerce')

df['quantity'] = df['quantity'].astype(int)

log\_data\_info(df, "Cleaned")

# Connect to PostgreSQL

engine = create\_engine(f'postgresql://{DB\_USER}:{DB\_PASSWORD}@{DB\_HOST}:{DB\_PORT}/{DB\_NAME}')

# Define schema

create\_table\_query = f"""

CREATE TABLE IF NOT EXISTS {TABLE\_NAME} (

    transaction\_id UUID PRIMARY KEY,

    customer\_id UUID,

    transaction\_amount NUMERIC,

    transaction\_date TIMESTAMP,

    payment\_method VARCHAR(50),

    product\_category VARCHAR(100),

    quantity INT,

    customer\_age INT,

    customer\_location VARCHAR(100),

    device\_used VARCHAR(50),

    ip\_address VARCHAR(50),

    shipping\_address TEXT,

    billing\_address TEXT,

    is\_fraudulent BOOLEAN,

    account\_age\_days INT,

    transaction\_hour INT

);

"""

# Execute schema creation

with engine.begin() as conn:

    conn.execute(text(create\_table\_query))

# Load data into PostgreSQL

df.to\_sql(TABLE\_NAME, engine, if\_exists='append', index=False)

print("Data successfully loaded into PostgreSQL.")

## Output

Raw Data Snapshot:

Transaction ID Customer ID Transaction Amount Transaction Date ... Billing Address Is Fraudulent Account Age Days Transaction Hour

0 15d2e414-8735-46fc-9e02-80b472b2580f d1b87f62-51b2-493b-ad6a-77e0fe13e785 58.09 2024-02-20 05:58:41 ... Unit 8934 Box 0058\nDPO AA 05437 0.0 30.0 5.0

1 0bfee1a0-6d5e-40da-a446-d04e73b1b177 37de64d5-e901-4a56-9ea0-af0c24c069cf 389.96 2024-02-25 08:09:45 ... 634 May Keys\nPort Cherylview, NV 75063 0.0 72.0 8.0

2 e588eef4-b754-468e-9d90-d0e0abfc1af0 1bac88d6-4b22-409a-a06b-425119c57225 134.19 2024-03-18 03:42:55 ... 16282 Dana Falls Suite 790\nRothhaven, IL 15564 0.0 63.0 3.0

3 4de46e52-60c3-49d9-be39-636681009789 2357c76e-9253-4ceb-b44e-ef4b71cb7d4d 226.17 2024-03-16 20:41:31 ... 828 Strong Loaf Apt. 646\nNew Joshua, UT 84798 0.0 124.0 20.0

4 074a76de-fe2d-443e-a00c-f044cdb68e21 45071bc5-9588-43ea-8093-023caec8ea1c 121.53 2024-01-15 05:08:17 ... 29799 Jason Hills Apt. 439\nWest Richardtown, ... 0.0 158.0 5.0

[5 rows x 16 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1472952 entries, 0 to 1472951

Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Transaction ID 1472950 non-null object

1 Customer ID 1472943 non-null object

2 Transaction Amount 1472949 non-null object

3 Transaction Date 1472951 non-null object

4 Payment Method 1472948 non-null object

5 Product Category 1472949 non-null object

6 Quantity 1472951 non-null float64

7 Customer Age 1472952 non-null object

8 Customer Location 1472949 non-null object

9 Device Used 1472949 non-null object

10 IP Address 1472949 non-null object

11 Shipping Address 1472951 non-null object

12 Billing Address 1472951 non-null object

13 Is Fraudulent 1472949 non-null float64

14 Account Age Days 1472948 non-null float64

15 Transaction Hour 1472945 non-null float64

dtypes: float64(4), object(12)

memory usage: 179.8+ MB

None

Missing values per column:

Transaction ID 2

Customer ID 9

Transaction Amount 3

Transaction Date 1

Payment Method 4

Product Category 3

Quantity 1

Customer Age 0

Customer Location 3

Device Used 3

IP Address 3

Shipping Address 1

Billing Address 1

Is Fraudulent 3

Account Age Days 4

Transaction Hour 7

dtype: int64

Cleaned Data Snapshot:

transaction\_id customer\_id transaction\_amount ... is\_fraudulent account\_age\_days transaction\_hour

0 15d2e414-8735-46fc-9e02-80b472b2580f d1b87f62-51b2-493b-ad6a-77e0fe13e785 58.09 ... 0 30 5

1 0bfee1a0-6d5e-40da-a446-d04e73b1b177 37de64d5-e901-4a56-9ea0-af0c24c069cf 389.96 ... 0 72 8

2 e588eef4-b754-468e-9d90-d0e0abfc1af0 1bac88d6-4b22-409a-a06b-425119c57225 134.19 ... 0 63 3

3 4de46e52-60c3-49d9-be39-636681009789 2357c76e-9253-4ceb-b44e-ef4b71cb7d4d 226.17 ... 0 124 20

4 074a76de-fe2d-443e-a00c-f044cdb68e21 45071bc5-9588-43ea-8093-023caec8ea1c 121.53 ... 0 158 5

[5 rows x 16 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1472952 entries, 0 to 1472951

Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 transaction\_id 1472952 non-null object

1 customer\_id 1472952 non-null object

2 transaction\_amount 1472952 non-null float64

3 transaction\_date 1472952 non-null datetime64[ns]

4 payment\_method 1472952 non-null object

5 product\_category 1472952 non-null object

6 quantity 1472952 non-null int64

7 customer\_age 1472952 non-null int64

8 customer\_location 1472952 non-null object

9 device\_used 1472952 non-null object

10 ip\_address 1472952 non-null object

11 shipping\_address 1472952 non-null object

12 billing\_address 1472952 non-null object

13 is\_fraudulent 1472952 non-null int64

14 account\_age\_days 1472952 non-null int64

15 transaction\_hour 1472952 non-null int64

dtypes: datetime64[ns](1), float64(1), int64(5), object(9)

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

13 is\_fraudulent 1472952 non-null int64

14 account\_age\_days 1472952 non-null int64

15 transaction\_hour 1472952 non-null int64

dtypes: datetime64[ns](1), float64(1), int64(5), object(9)

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

product\_category 0

quantity 0

customer\_age 0

customer\_location 0

13 is\_fraudulent 1472952 non-null int64

14 account\_age\_days 1472952 non-null int64

15 transaction\_hour 1472952 non-null int64

dtypes: datetime64[ns](1), float64(1), int64(5), object(9)

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

product\_category 0

quantity 0

customer\_age 0

13 is\_fraudulent 1472952 non-null int64

14 account\_age\_days 1472952 non-null int64

15 transaction\_hour 1472952 non-null int64

dtypes: datetime64[ns](1), float64(1), int64(5), object(9)

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

14 account\_age\_days 1472952 non-null int64

15 transaction\_hour 1472952 non-null int64

dtypes: datetime64[ns](1), float64(1), int64(5), object(9)

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

memory usage: 179.8+ MB

None

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

Missing values per column:

transaction\_id 0

customer\_id 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

product\_category 0

transaction\_amount 0

transaction\_date 0

payment\_method 0

product\_category 0

quantity 0

transaction\_date 0

payment\_method 0

product\_category 0

quantity 0

customer\_age 0

customer\_location 0

device\_used 0

ip\_address 0

product\_category 0

quantity 0

customer\_age 0

customer\_location 0

device\_used 0

ip\_address 0

quantity 0

customer\_age 0

customer\_location 0

device\_used 0

ip\_address 0

shipping\_address 0

customer\_age 0

customer\_location 0

device\_used 0

ip\_address 0

shipping\_address 0

device\_used 0

ip\_address 0

shipping\_address 0

shipping\_address 0

billing\_address 0

is\_fraudulent 0

account\_age\_days 0

transaction\_hour 0

billing\_address 0

is\_fraudulent 0

account\_age\_days 0

transaction\_hour 0

is\_fraudulent 0

account\_age\_days 0

transaction\_hour 0

dtype: int64

account\_age\_days 0

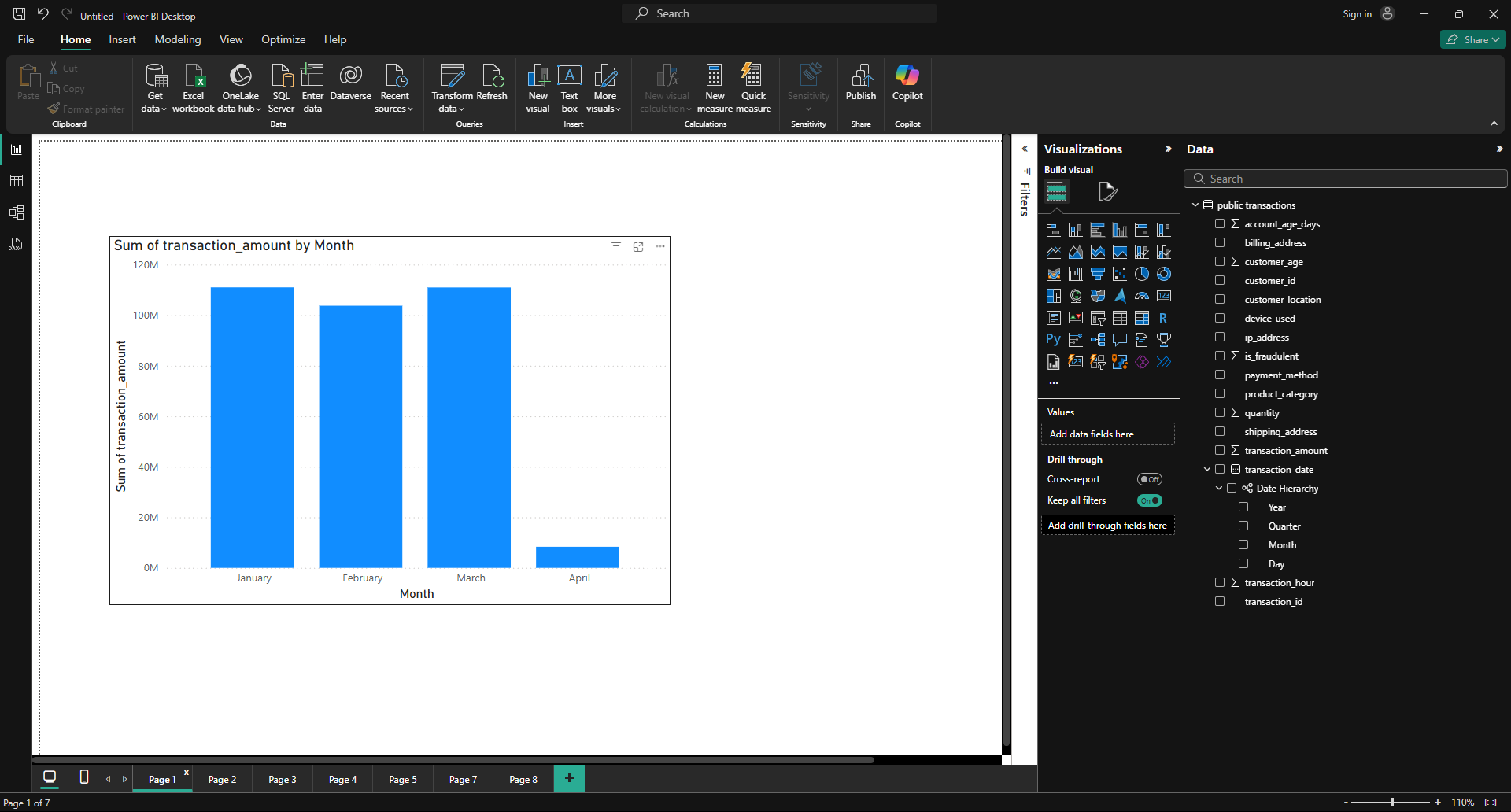
transaction\_hour 0

dtype: int64

dtype: int64

Data successfully loaded into PostgreSQL.

## Data Visualization and Insights: Transaction and Customer Analysis



### Description of the Bar Chart:

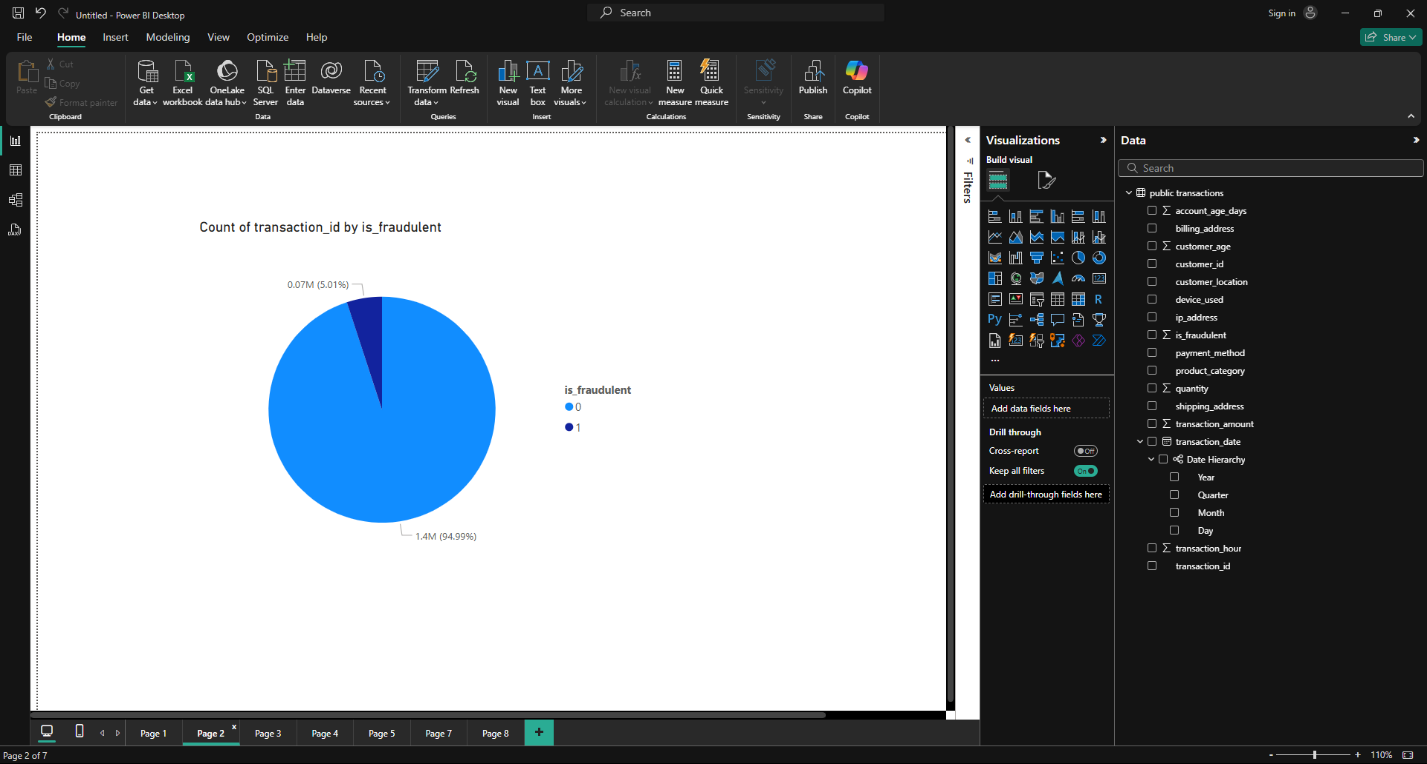
* represents the total transaction amount for each month, allowing for a comparison of transaction volume across different months.

**Axes:**

* **X-axis (Horizontal):** This axis represents the **Month**. The chart shows four months: January, February, March, and April.
* **Y-axis (Vertical):** This axis represents the **Sum of transaction amount**. The values on this axis indicate the total amount of transactions for each corresponding month. The axis is scaled in millions (M), with increments of 20 million (20M, 40M, 60M, etc.).

**Information Conveyed by the Chart:**

* **Trend Over Time:** The chart reveals the trend of transaction amounts over the four months. We can observe that the transaction amount is relatively high and consistent in January, February, and March. However, there's a significant drop in April.
* **Comparison of Months:** The chart facilitates a direct comparison of transaction amounts between the different months. It's immediately clear that April has a substantially lower transaction volume compared to the preceding months.
* **Potential Insights/Questions:** The sharp decline in April raises questions and warrants further investigation. Possible reasons for this drop could include:
  + Seasonal factors affecting business.
  + A change in business operations or strategy.
  + Data errors or incomplete data for April.
  + Specific events or promotions that boosted transactions in the earlier months.



### Description of the Pie Chart:

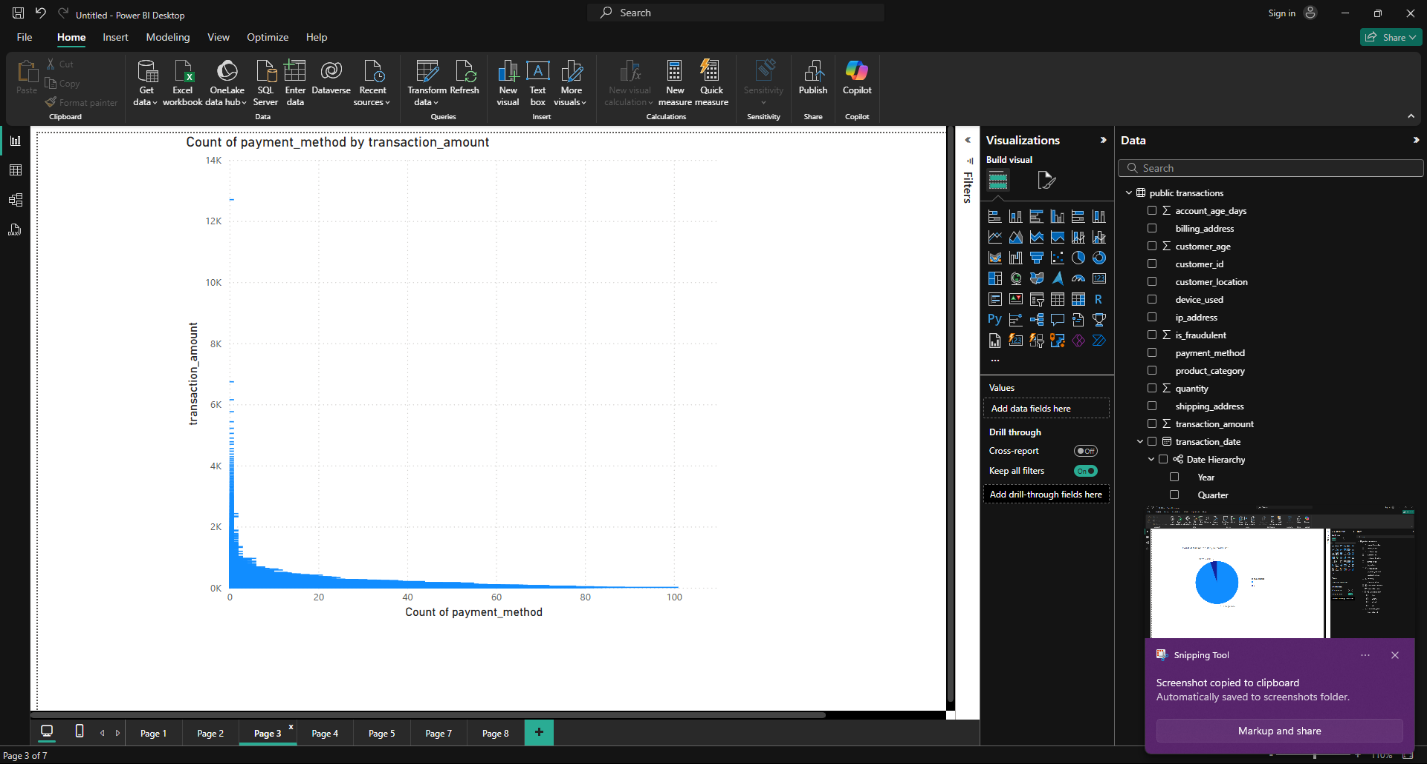
This visualization is a **pie chart titled "Count of transaction\_id by is\_fraudulent"**. It displays the proportion of transactions that are flagged as fraudulent versus those that are not.

**Elements and Labels:**

* **Pie Slices:** The pie chart is divided into two slices:
  + One slice represents the count of transactions where "is\_fraudulent" is likely true (indicated by the value "1").
  + The other slice represents the count of transactions where "is\_fraudulent" is likely false (indicated by the value "0").
* **Labels:** Each slice is labeled with:
  + The value of the "is\_fraudulent" field (0 or 1).
  + The count of transactions in that category (e.g., 1.84M).
  + The percentage of the total transactions that the slice represents.

**Information Conveyed by the Chart:**

* **Proportion of Fraudulent Transactions:** The primary insight conveyed is the proportion of transactions flagged as fraudulent compared to the total number of transactions. The chart clearly shows that the vast majority of transactions are *not* flagged as fraudulent (1.84M, representing 99.96% of total transactions).
* **Very Low Fraud Rate:** The visualization highlights a very low fraud rate (0.04%) based on the "is\_fraudulent" field. Only a tiny fraction of transactions (741) are flagged as potentially fraudulent.
* **Potential for Further Investigation:** While the chart shows a low fraud rate, even a small number of fraudulent transactions can be significant. This visualization would likely prompt further investigation into the characteristics of those 741 transactions to understand the patterns or causes of potential fraud.



### Description of the Chart:

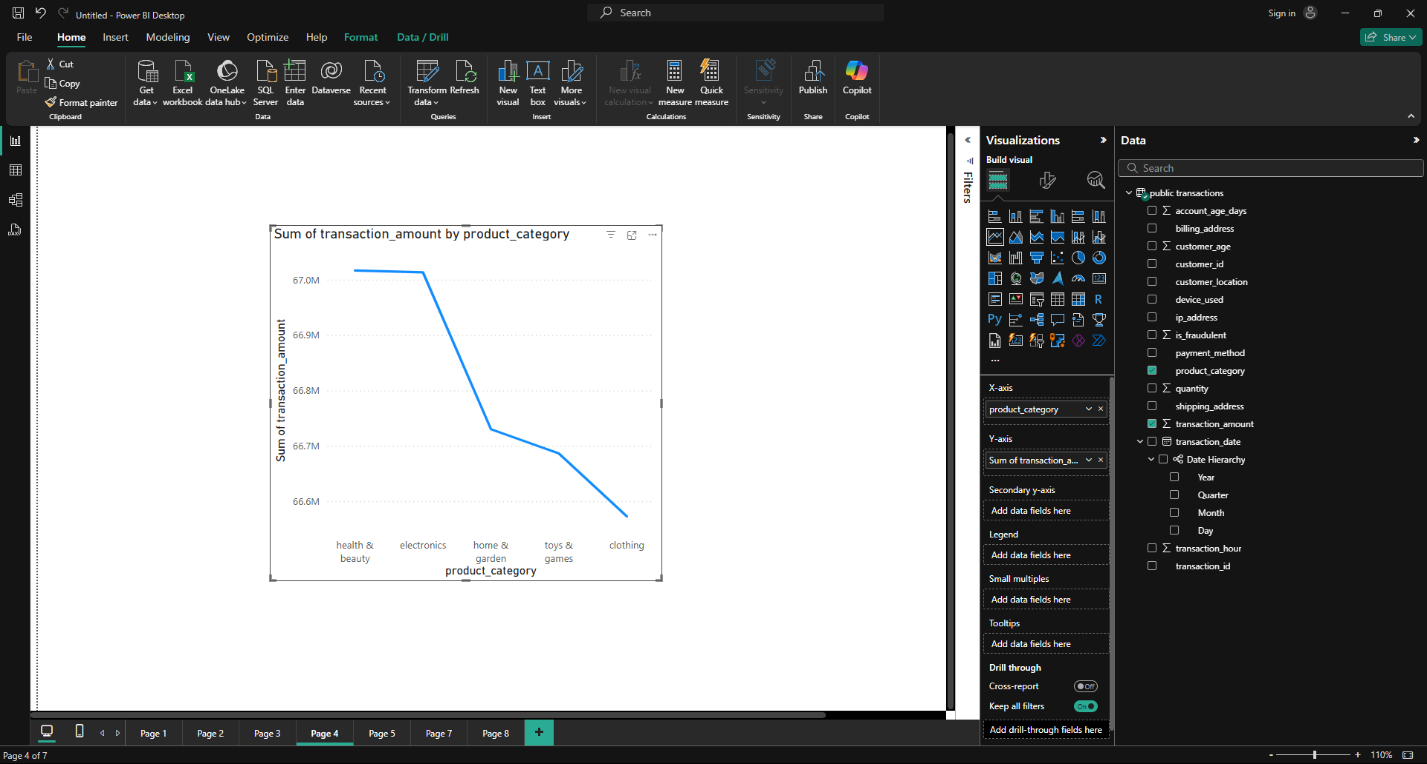
This visualization is a **column chart (or bar chart if oriented horizontally) titled "Count of payment method by transaction amount"**. It shows the distribution of transaction amounts, categorized by the payment method used.

**Axes:**

* **X-axis (Horizontal):** This axis represents the **Count of payment method**. It appears to be showing the number of times each payment method was used, but the label is a bit unclear. It might be better phrased as "Count of Transactions" or "Transaction Frequency".
* **Y-axis (Vertical):** This axis represents the **transaction amount**. It shows the range of transaction amounts, likely divided into bins or groups.

**Elements and Labels:**

* **Columns (or Bars):** Each column represents a specific range of transaction amounts. The height of the column corresponds to the number of times that payment method was used for transactions within that amount range.
* **Tooltips:** (Not explicitly visible but likely present) When hovering over a column, Power BI would typically display a tooltip showing the exact transaction amount range and the count of transactions (or frequency) for that range and payment method.



### Description of the Line Chart:

This visualization is a **line chart titled "Sum of transaction amount by product category"**. It displays the trend of total transaction amounts for different product categories.

**Axes:**

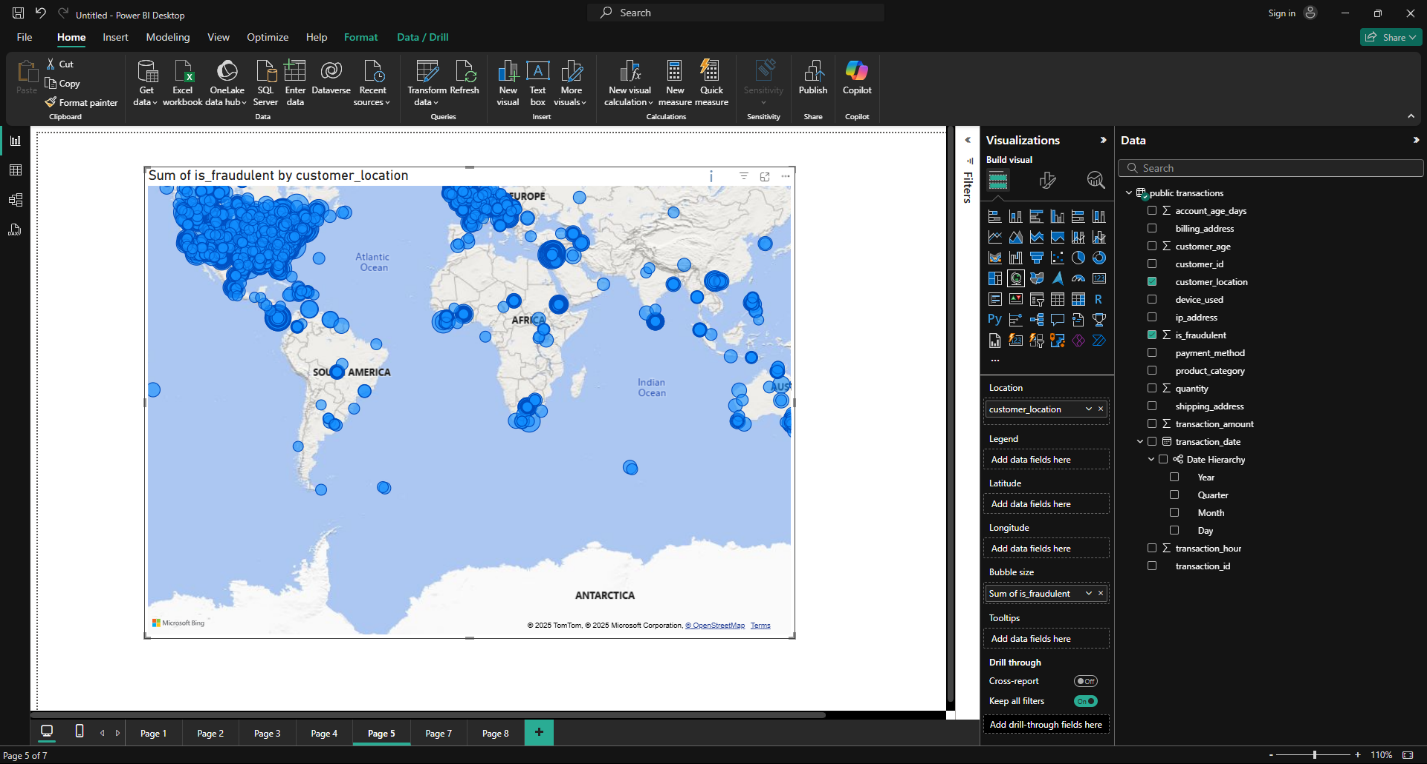
* **X-axis (Horizontal):** This axis represents the **product category**. The chart shows five categories: health & beauty, electronics, home & garden, clothing, and games.
* **Y-axis (Vertical):** This axis represents the **Sum of transaction amount**. The values on this axis indicate the total amount of transactions for each corresponding product category. The axis is scaled in millions (M), with increments of 20 million (20M, 40M, 60M, etc.).

**Elements and Labels:**

* **Line:** The line connects the data points for each product category, visually showing the trend of transaction amounts across categories.
* **Data Points:** Each point on the line represents the total transaction amount for a specific product category.

**Information Conveyed by the Chart:**

* **Comparison of Product Category Performance:** The chart allows for a direct comparison of the total transaction amounts generated by different product categories.
* **Relative Ranking of Categories:** It's easy to see the relative ranking of the categories based on their transaction amounts. Electronics stands out as the highest, followed by home & garden, then health & beauty, with clothing and games at the lower end.
* **Potential Insights/Questions:**
  + The significant difference in transaction amounts between categories suggests varying levels of demand or market size for these product types.
  + The high performance of electronics raises questions about the factors driving its sales (e.g., pricing, promotions, seasonality).
  + The lower transaction amounts for clothing and games might prompt further investigation into potential strategies to boost sales in these categories.



### Description of the Map Chart:

This visualization is a **filled map chart titled "Sum of is\_fraudulent by customer location"**. It aims to show the geographic distribution of potentially fraudulent transactions based on customer location.

**Elements and Labels:**

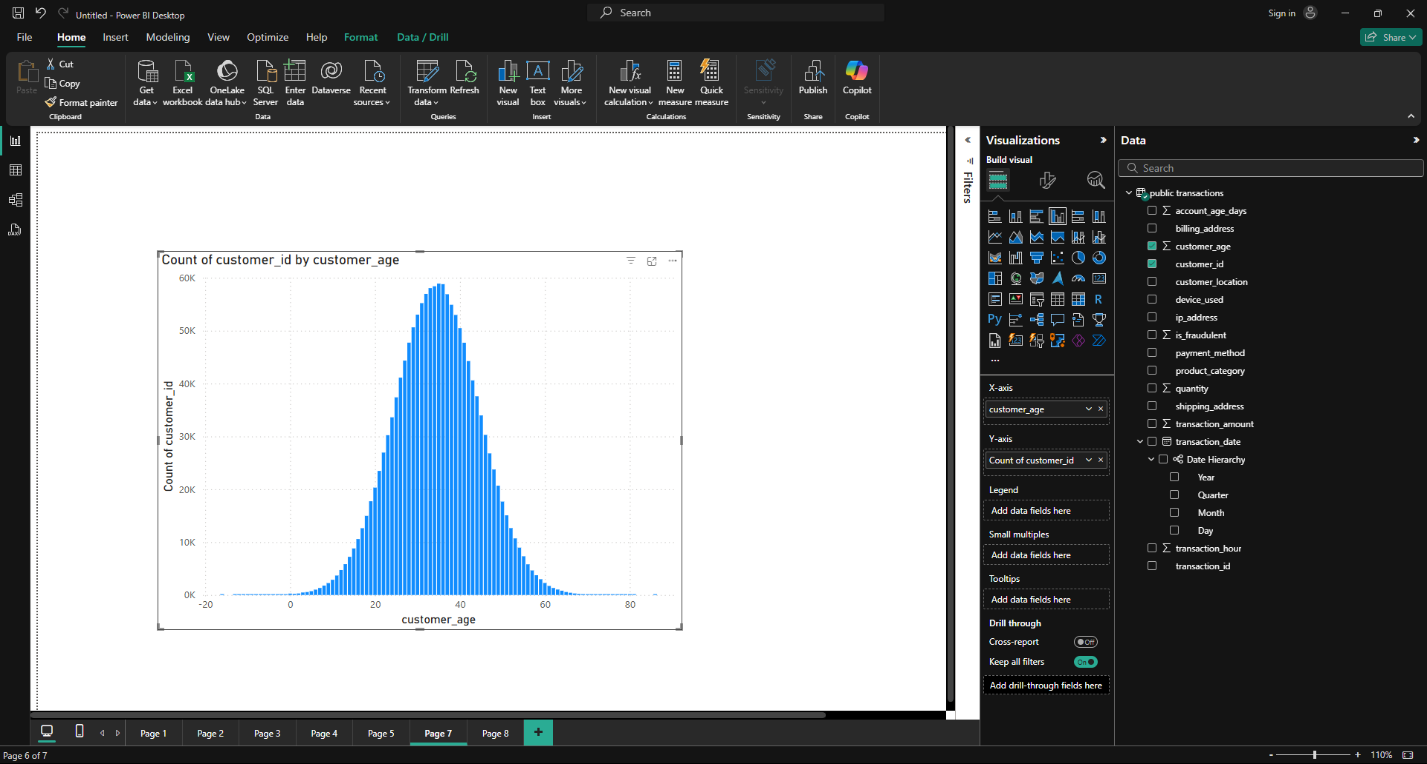
* **Map:** A world map is displayed, with country/region boundaries outlined.
* **Filled Regions:** Countries or regions with potentially fraudulent transactions are filled with a blue color. The intensity of the blue likely corresponds to the number or proportion of fraudulent transactions in that area.
* **Bubbles:** (Potentially, but not clearly visible) There might be bubbles on top of filled regions, with the size of the bubble indicating the magnitude of fraudulent transactions. However, this is difficult to confirm from the image alone.
* **Title:** The title "Sum of is\_fraudulent by customer location" indicates that the visualization is aggregating the "is\_fraudulent" field (likely a binary flag) by customer location.
* **Scale:** The map includes a scale at the bottom left, likely indicating the range of values represented by the color intensity or bubble size.

**Information Conveyed by the Chart:**

* **Geographic Distribution of Fraud:** The primary insight is the geographic distribution of potentially fraudulent activities. The filled regions highlight areas with a higher concentration of "is\_fraudulent" transactions.
* **Identification of Hotspots:** The visualization helps identify potential "hotspots" or regions with unusually high rates of fraudulent transactions.
* **Global Patterns:** The map allows for the observation of global patterns and trends in fraudulent activities.

**Potential Issues and Recommendations:**

* **Clarity of Bubbles:** If bubbles are intended to be part of the visualization, they are not clearly visible in the image. Clearer bubble representation would enhance the understanding of the magnitude of fraud in different locations.
* **Color Scale:** The color scale's range and meaning should be easily understandable. It should clearly show how the color intensity relates to the number or proportion of fraudulent transactions.
* **Tooltips:** Interactive tooltips would be beneficial. Hovering over a region should display specific details, such as the region name, the count of fraudulent transactions, and any relevant metrics.
* **Drill-down Functionality:** Ideally, the map should allow for drill-down functionality. Clicking on a region could zoom in and provide more granular data for that specific area.



### Description of the Column Chart:

This visualization is a **column chart (or bar chart if oriented horizontally) titled "Count of customer\_id by customer\_age"**. It displays the distribution of customers across different age groups.

**Axes:**

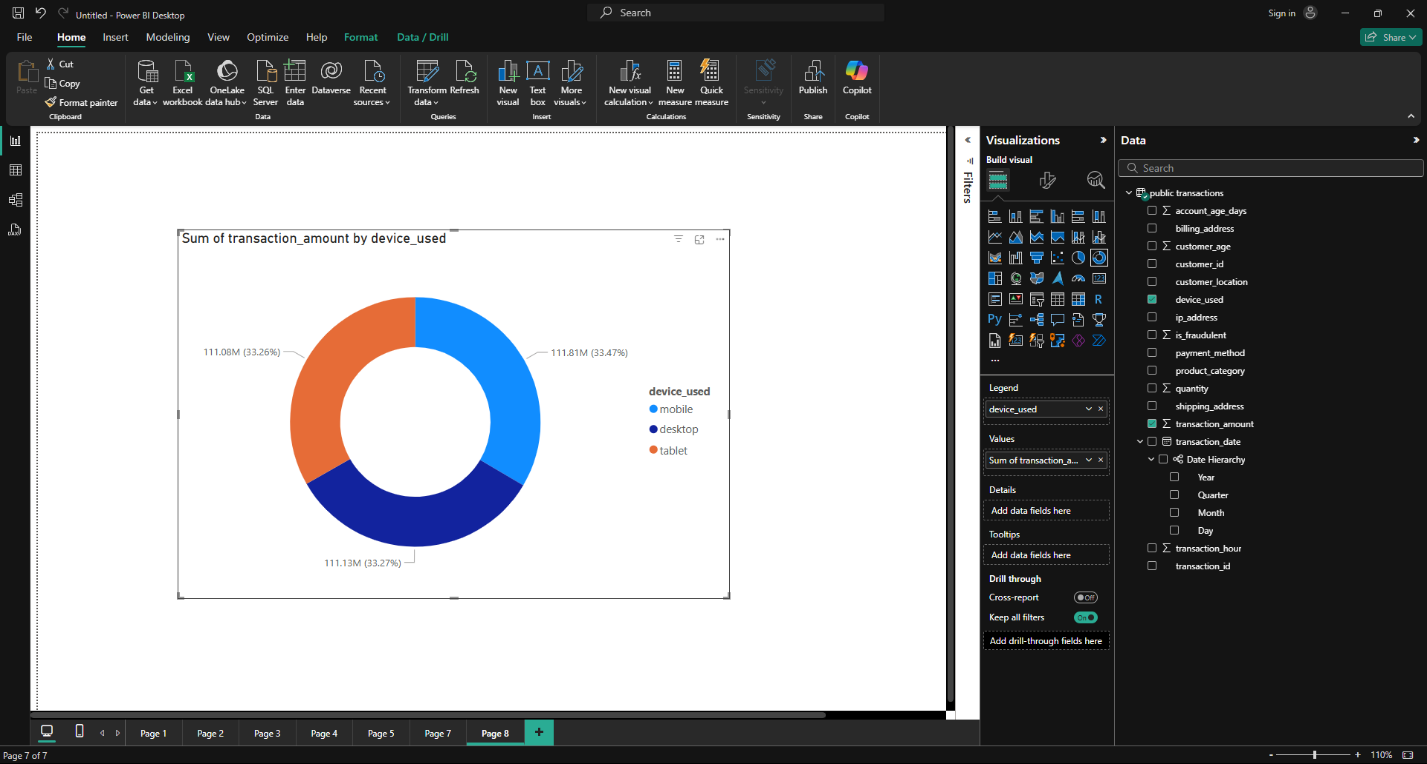
* **X-axis (Horizontal):** This axis represents the **customer age**. The chart shows a range of ages, likely grouped into bins (e.g., 0-10, 10-20, etc.) or individual ages if the data allows.
* **Y-axis (Vertical):** This axis represents the **Count of customer\_id**. It shows the number of customers within each age group or at each specific age.

**Elements and Labels:**

* **Columns (or Bars):** Each column represents a specific age group or individual age. The height of the column corresponds to the number of customers in that age group or at that age.
* **Title:** The title "Count of customer\_id by customer\_age" clearly indicates the chart's purpose.

**Information Conveyed by the Chart:**

* **Customer Age Distribution:** The chart provides a clear picture of how customers are distributed across different age groups.
* **Identification of Peak Age Groups:** The chart helps identify age groups with the highest concentration of customers. In this case, there appears to be a bell-shaped curve, suggesting a normal distribution with a peak in the middle age ranges.
* **Potential Insights/Questions:**
  + The concentration of customers in specific age groups might inform marketing strategies and product development.
  + Deviations from a normal distribution could indicate specific customer segments or trends.



### Description of the Donut Chart:

This visualization is a **donut chart titled "Sum of transaction amount by device used"**. It shows the proportion of total transaction amounts attributed to different device types.

**Elements and Labels:**

* **Donut Shape:** The chart is circular with a hole in the center, resembling a donut.
* **Slices:** The donut is divided into slices, each representing a different device type.
* **Labels:** Each slice is labeled with:
  + The device type (mobile, desktop, tablet).
  + The transaction amount associated with that device type (e.g., 11.13M).
  + The percentage of the total transaction amount that the slice represents (e.g., 3.27%).

**Information Conveyed by the Chart:**

* **Device Usage for Transactions:** The chart clearly shows the distribution of transaction amounts across different devices.
* **Dominant Device:** It's immediately evident that desktop transactions account for the largest share of the total transaction amount (111.41M, 96.67%).
* **Smaller Contributions:** Mobile and tablet transactions make up a much smaller portion of the total transaction amount.
* **Relative Comparison:** The chart allows for a quick visual comparison of the relative contributions of each device type to the overall transaction volume

# Conclusion

This script efficiently processes e-commerce transaction data, cleans it, and stores it in a PostgreSQL database for analysis. It ensures data integrity by handling duplicates and missing values while structuring the database schema for efficient querying.

**Overall Insights from the Power BI Report:**

The visualizations collectively provide a snapshot of various aspects of transaction data, customer demographics, and potential fraud.

* **Transaction Trends:**
  + Transaction amounts were relatively stable in January, February, and March but dropped significantly in April (bar chart).
  + Electronics is the highest-performing product category in terms of transaction amount, followed by home & garden (line chart).
  + Desktop transactions dominate in terms of total transaction amount, with mobile and tablet transactions contributing much less (donut chart).
* **Customer Demographics:**
  + Customer age distribution appears to follow a bell-shaped curve, with a concentration in the middle age ranges (column chart).
* **Fraud Analysis:**
  + The overall fraud rate is very low (0.04%) based on the "is\_fraudulent" field (pie chart).
  + The filled map chart aims to show the geographic distribution of potentially fraudulent transactions, but its effectiveness is limited by unclear visuals and lack of interactivity.

**Key Takeaways:**

* The report highlights a significant drop in transaction amount in April, warranting further investigation into potential causes.
* Electronics stands out as a high-performing product category, while clothing and games may require strategies to boost sales.
* The dominance of desktop transactions suggests a need to optimize mobile and tablet experiences to drive more transactions on those platforms.
* While the overall fraud rate is low, further analysis of the flagged transactions is necessary to understand fraud patterns and mitigate risks.
* The visualizations demonstrate Power BI's ability to provide insights into various aspects of business data, but there's room for improvement in terms of visual clarity, interactivity, and labeling to enhance their effectiveness.

**Recommendations:**

* Investigate the reasons for the April transaction decline.
* Explore strategies to boost sales in underperforming product categories.
* Optimize mobile and tablet experiences to increase transaction volume on those platforms.
* Conduct further analysis of potentially fraudulent transactions to identify patterns and mitigate risks.
* Improve the visual clarity, interactivity, and labeling of the visualizations to enhance their effectiveness.