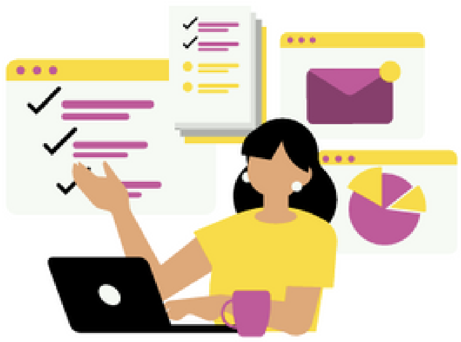
Fradulent Transaction Detection

MySQL Project

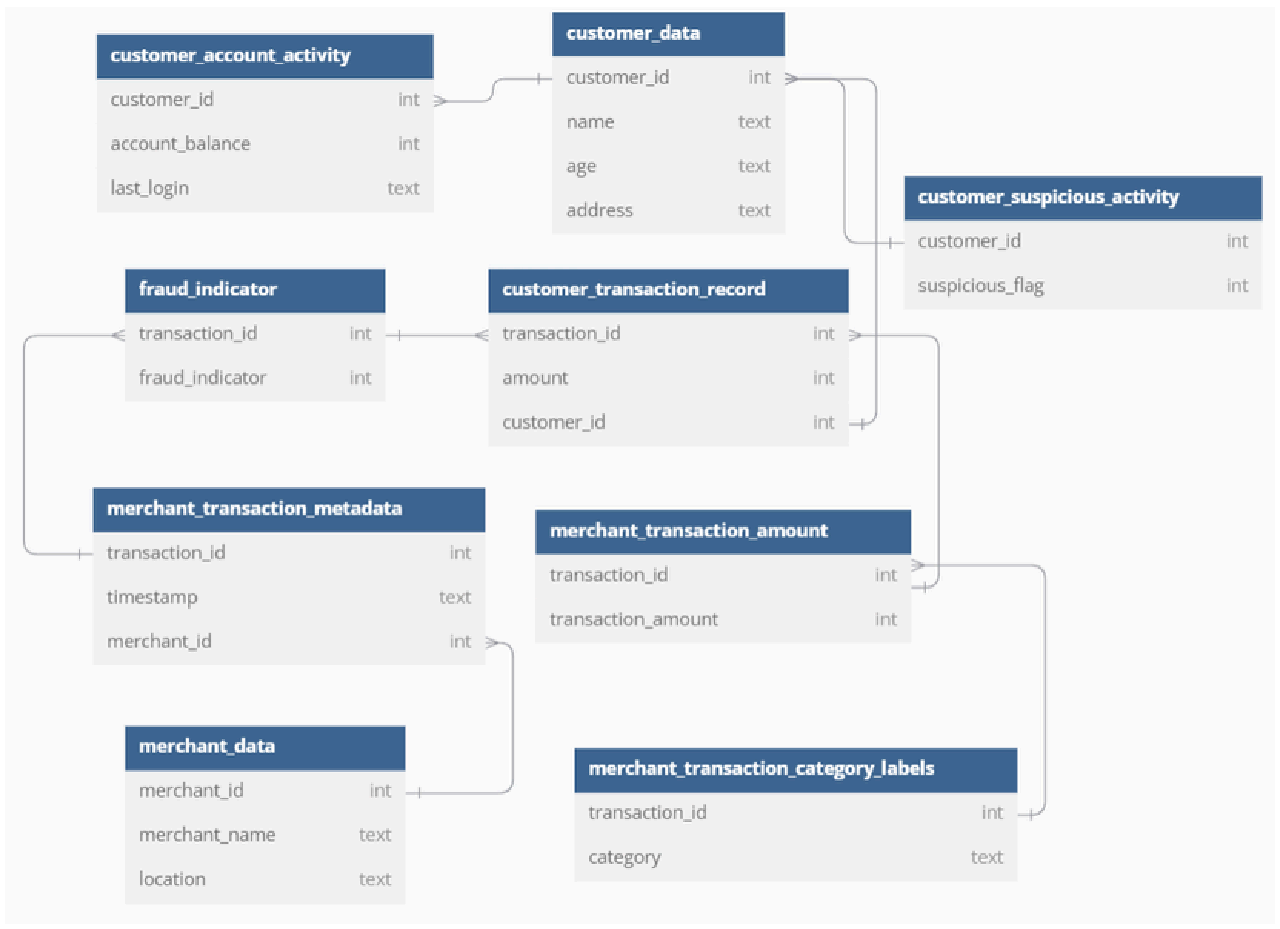
Objective:

- The primary objective is to detect payment fraud through real-time monitoring of transactions and various metrics.

- We need to examine dataset with SQL and help understand past customers transactions and suggest new risk rules.



Fraudulent Transaction Detection Database



By Vedant Burande

Fradulent Transaction Detection

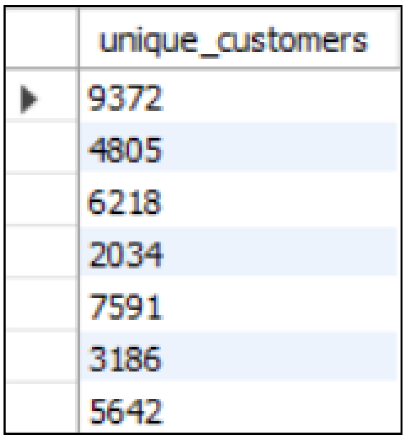
MySQL Project

Q1. How many Unique Customers are there in the dataset?

- Input:

```
SELECT COUNT(DISTINCT customer_id) AS unique_customers
FROM customer_data;
SELECT DISTINCT customer_id AS unique_customers
FROM customer_data;
```

- Output:



```
unique_customers
9372
4805
6218
2034
7591
3186
5642
```

Q2. Which customer have the highest & lowest account balance?

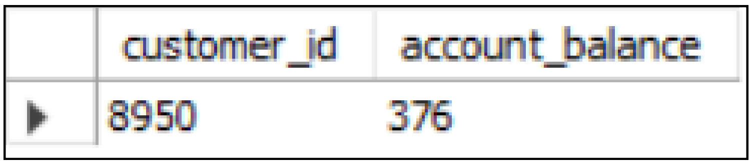
- Input:

```
SELECT customer_id, account_balance
FROM customer_account_activity
WHERE account_balance = (SELECT max(account_balance) FROM
customer_account_activity);
SELECT customer_id, account_balance
FROM customer_account_activity
WHERE account_balance = (SELECT min(account_balance) FROM
customer_account_activity);
```

By Vedant Burande

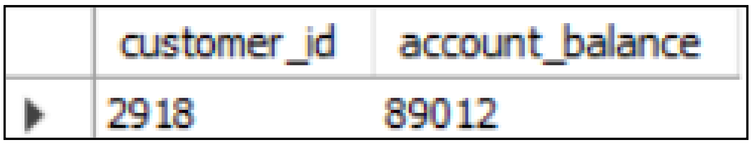
- Output:

<u>Lowest-</u>



	customer_id	account_balance
▶	8950	376

<u>Highest-</u>



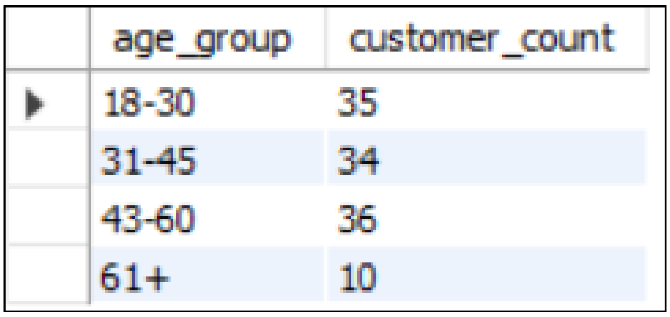
	customer_id	account_balance
▶	2918	89012

Q3. What is the distribution of customer ages in the dataset?

- Input:

```
SELECT
 CASE  WHEN age between 18 AND 30 THEN '18-30'
          WHEN age between 31 AND 45 THEN '31-45'
          WHEN age between 46 AND 60 THEN '43-60'
 ELSE '61+'
 END AS age_group, COUNT(*) AS customer_count FROM customer_data
GROUP BY age_group ORDER BY age_group;
```

- Output:



	age_group	customer_count
▶	18-30	35
	31-45	34
	43-60	36
	61+	10

Fradulent Transaction Detection

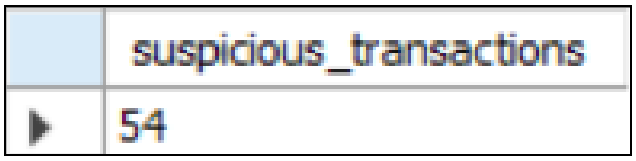
MySQL Project

Q4. How many customer are engaged in suspicious activity?

- Input:

```
SELECT COUNT(DISTINCT customer_id) AS suspicious_transactions
FROM customer_suspisious_activity
WHERE suspisious_flag = '1';
```

- Output:



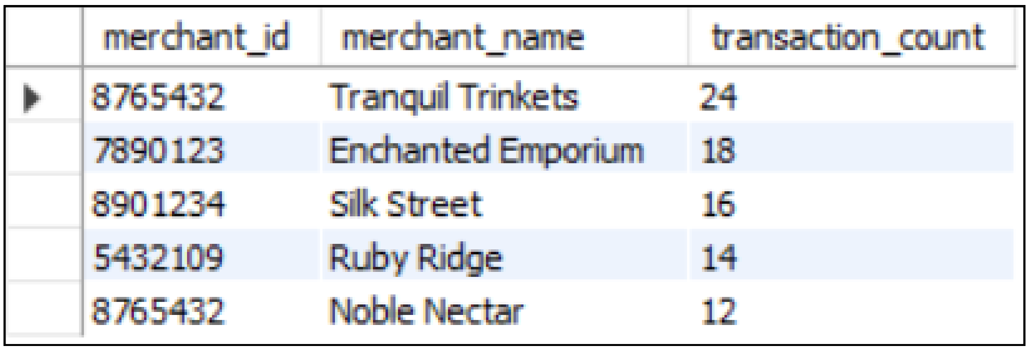
suspicious_transactions
54

Q5. Top 5 merchants who have the Highest number of Transaction

- Input:

```
SELECT md.merchant_id, md.merchant_name, COUNT(*) AS
transaction_count
FROM merchant_data md
INNER JOIN merchant_transaction_metadata mtm
USING(merchant_id) GROUP BY merchant_id, merchant_name
ORDER BY transaction_count DESC LIMIT 5;
```

- Output:



merchant_id	merchant_name	transaction_count
8765432	Tranquil Trinkets	24
7890123	Enchanted Emporium	18
8901234	Silk Street	16
5432109	Ruby Ridge	14
8765432	Noble Nectar	12

By Vedant Burande

Fradulent Transaction Detection

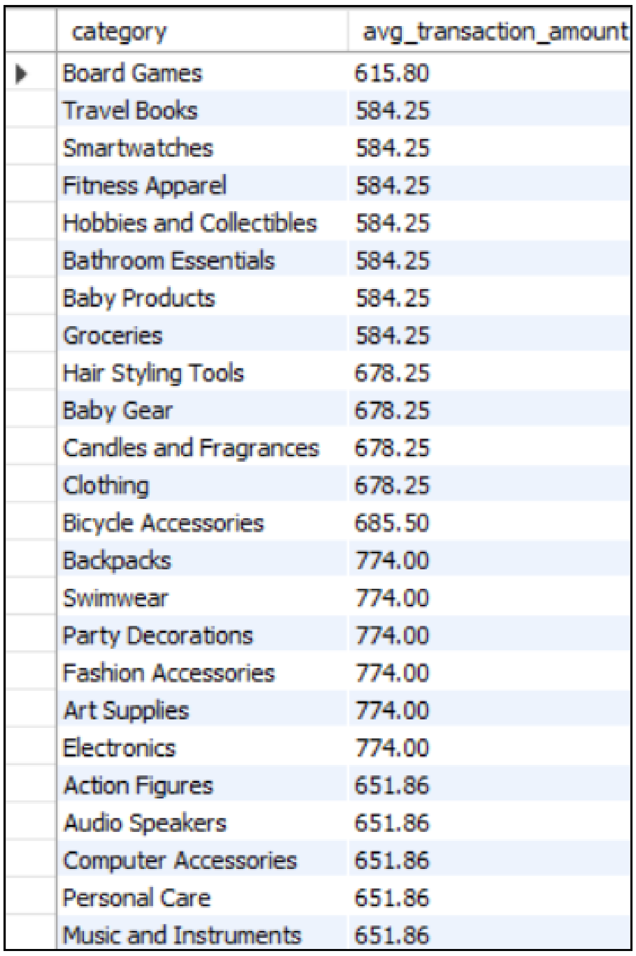
MySQL Project

Q6. What is the average transaction amount for each merchants transaction category?

- Input:

```
SELECT category, round(avg(transaction_amount),2) AS
avg_transaction_amount
FROM merchant_transaction_amount
INNER JOIN merchant_transaction_category_labels
USING(transaction_id)
GROUP BY category;
```

- Output:



category	avg_transaction_amount
Board Games	615.80
Travel Books	584.25
Smartwatches	584.25
Fitness Apparel	584.25
Hobbies and Collectibles	584.25
Bathroom Essentials	584.25
Baby Products	584.25
Groceries	584.25
Hair Styling Tools	678.25
Baby Gear	678.25
Candles and Fragrances	678.25
Clothing	678.25
Bicycle Accessories	685.50
Backpacks	774.00
Swimwear	774.00
Party Decorations	774.00
Fashion Accessories	774.00
Art Supplies	774.00
Electronics	774.00
Action Figures	651.86
Audio Speakers	651.86
Computer Accessories	651.86
Personal Care	651.86
Music and Instruments	651.86

By Vedant Burande

Fradulent Transaction Detection

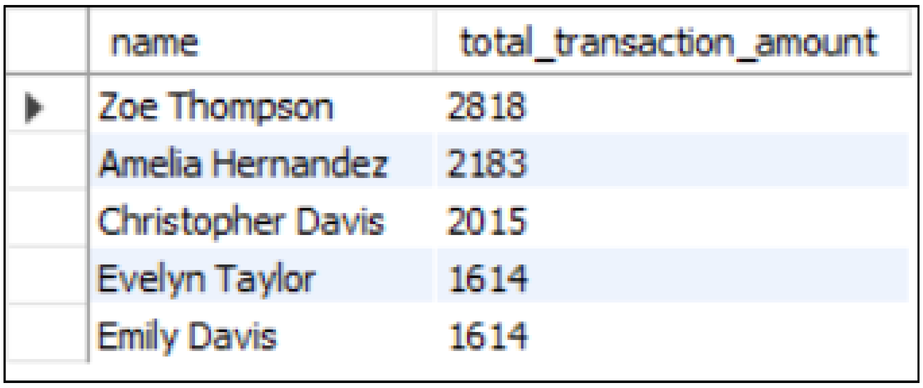
MySQL Project

Q7. Top 5 customers having the highest total transaction amounts.

- Input:

```
SELECT cd.name, round(sum(amount),2) AS total_transaction_amount
FROM customer_data cd
INNER JOIN customer_transaction_records ctr USING (customer_id)
GROUP BY cd.name ORDER BY total_transaction_amount DESC LIMIT 5;
```

- Output:



name	total_transaction_amount
Zoe Thompson	2818
Amelia Hernandez	2183
Christopher Davis	2015
Evelyn Taylor	1614
Emily Davis	1614

Q8. Which merchant have been associated with fradulent transactions?

- Input:

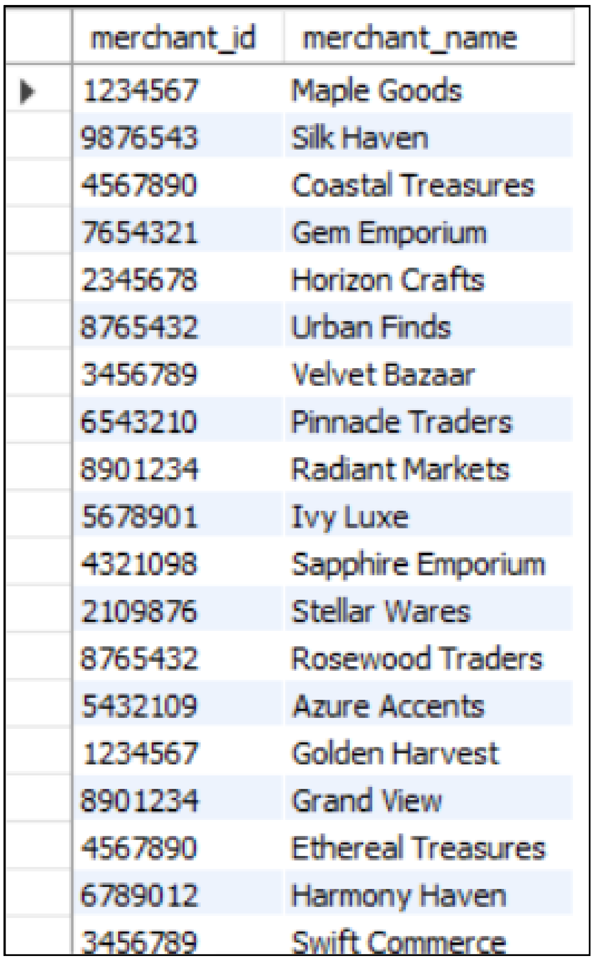
```
SELECT DISTINCT md.merchant_id, md.merchant_name
FROM merchant_data md
INNER JOIN merchant_transaction_metadata mtm USING
(merchant_id)
INNER JOIN fraud_indicator fi USING (transaction_id)
WHERE fi.fraud_indicator = 1;
```

By Vedant Burande

Fradulent Transaction Detection

MySQL Project

- Output:



merchant_id	merchant_name
1234567	Maple Goods
9876543	Silk Haven
4567890	Coastal Treasures
7654321	Gem Emporium
2345678	Horizon Crafts
8765432	Urban Finds
3456789	Velvet Bazaar
6543210	Pinnacle Traders
8901234	Radiant Markets
5678901	Ivy Luxe
4321098	Sapphire Emporium
2109876	Stellar Wares
8765432	Rosewood Traders
5432109	Azure Accents
1234567	Golden Harvest
8901234	Grand View
4567890	Ethereal Treasures
6789012	Harmony Haven
3456789	Swift Commerce

Q9. How many fradulent transactions have occured in each category?

- Input:

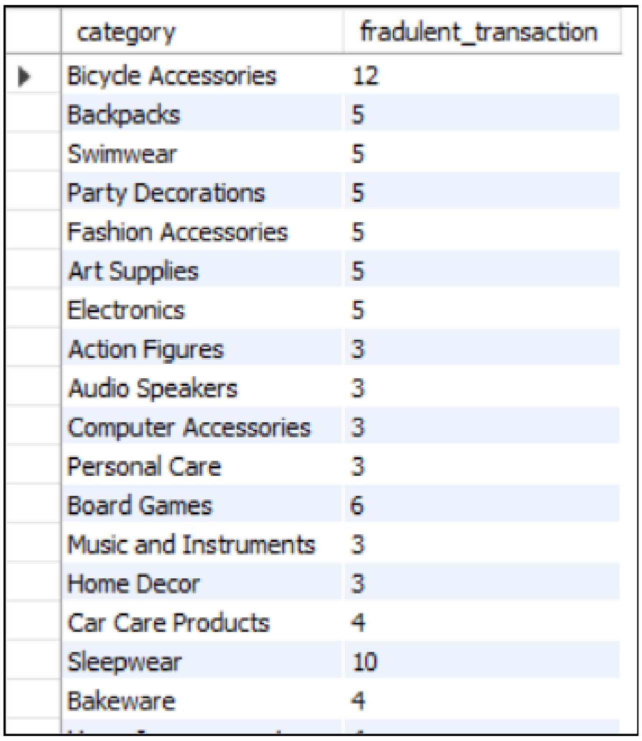
```
SELECT category, COUNT(*) AS fradulent_transaction
FROM merchant_transaction_category_labels
INNER JOIN fraud_indicator fi USING (transaction_id)
WHERE fi.fraud_indicator = 1 GROUP BY category;
```

- Output:

By Vedant Burande

Fradulent Transaction Detection

MySQL Project



category	fradulent_transaction
Bicycle Accessories	12
Backpacks	5
Swimwear	5
Party Decorations	5
Fashion Accessories	5
Art Supplies	5
Electronics	5
Action Figures	3
Audio Speakers	3
Computer Accessories	3
Personal Care	3
Board Games	6
Music and Instruments	3
Home Decor	3
Car Care Products	4
Sleepwear	10
Bakeware	4

Q10. Find the customers who have made transactions at multiple merchants AND display their names AND the number of unique merchants they have transacted with.

- Input:

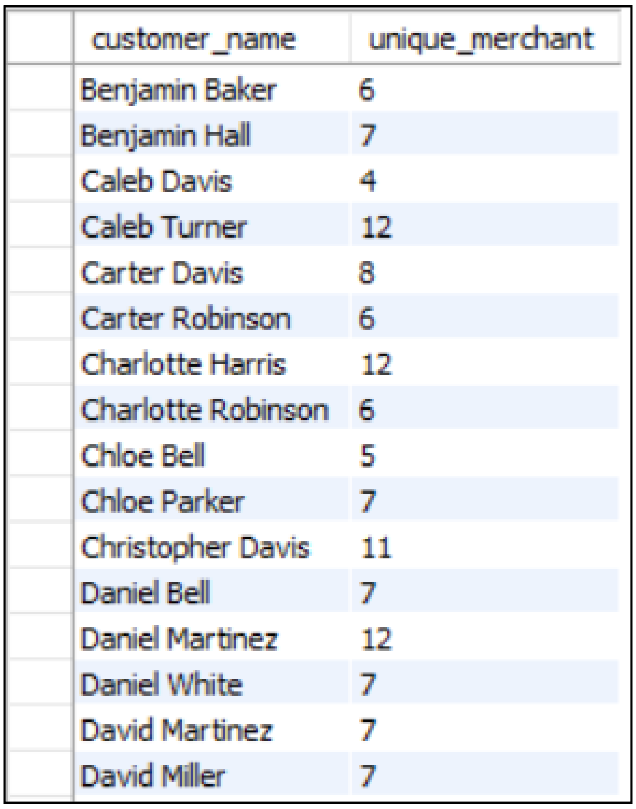
```
SELECT cd.name AS customer_name, COUNT(DISTINCT
md.merchant_id) AS unique_merchant
FROM customer_data cd INNER JOIN
customer_transaction_records ctr
USING (customer_id)
INNER JOIN merchant_transaction_metadata mtm USING
(transaction_id)
INNER JOIN merchant_data md USING (merchant_id)
GROUP BY cd.name having unique_merchant >1;
```

- Output:

By Vedant Burande

Fradulent Transaction Detection

MySQL Project



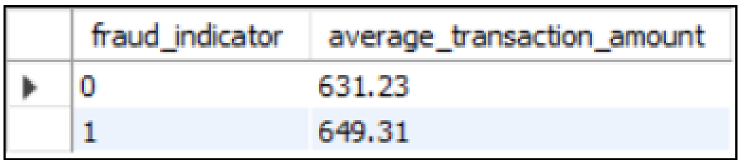
customer_name	unique_merchant
Benjamin Baker	6
Benjamin Hall	7
Caleb Davis	4
Caleb Turner	12
Carter Davis	8
Carter Robinson	6
Charlotte Harris	12
Charlotte Robinson	6
Chloe Bell	5
Chloe Parker	7
Christopher Davis	11
Daniel Bell	7
Daniel Martinez	12
Daniel White	7
David Martinez	7
David Miller	7

Q11. What is the average transaction amount for fraudulent transactions compared to non-fraudulent transactions?

- Input:

```
SELECT fi.fraud_indicator, round(avg(amount),2) AS
average_transaction_amount
FROM fraud_indicator fi
INNER JOIN customer_transaction_records ctr USING (transaction_id)
GROUP BY fraud_indicator;
```

- Output:



fraud_indicator	average_transaction_amount
0	631.23
1	649.31

By Vedant Burande

Fradulent Transaction Detection

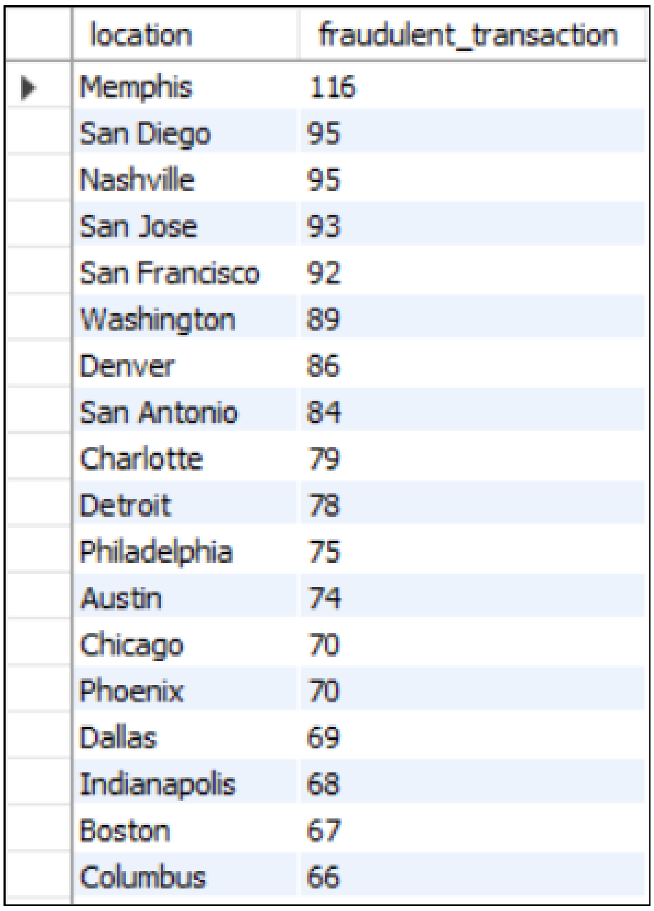
MySQL Project

Q12. Are there any regional patterns in fraudulent transactions?

- Input:

```
SELECT md.location, COUNT(*) AS fraudulent_transaction
FROM merchant_data md
INNER JOIN merchant_transaction_metadata mtm USING (merchant_id)
INNER JOIN fraud_indicator fi USING (transaction_id)
WHERE fraud_indicator = 1
GROUP BY md.location
ORDER BY fraudulent_transaction DESC;
```

- Output:



location	fraudulent_transaction
Memphis	116
San Diego	95
Nashville	95
San Jose	93
San Francisco	92
Washington	89
Denver	86
San Antonio	84
Charlotte	79
Detroit	78
Philadelphia	75
Austin	74
Chicago	70
Phoenix	70
Dallas	69
Indianapolis	68
Boston	67
Columbus	66

By Vedant Burande

Fradulent Transaction Detection

MySQL Project

Project Resources:

- CSV Files Dataset Link: https://drive.google.com/
- SQL Database Link: https://vedantburande.github.io/
- GitHub Project Link: https://vedantburande.github.io/

Profile:

- Portfolio: https://vedantburande.github.io/
- LinkedIn: https://www.linkedin.com/in/vedantburande/
- GitHub: https://github.com/VedantBurande
- Contact: **+91 97668 02199**

By Vedant Burande