context of the work

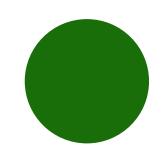
XYZ bank opened a large number of accounts across the globe: They have great cash flow and large amounts of transactions are happening

There are a huge number of transactions happening in the bank which raises suspicion about some accounts

They are evaluating the situation with the help of SQl finding suspicious accounts, the amount, trend, fraud dictation

This document contains a summary of SQL findings which were based on 1000 account reviews about fraud and the major details of accounts

Find the total transaction amount for all fraudulent transactions.



```
select sum(transaction_amount) as fraudAmount
from fraud_dection
where is_fraud = 1;
sum is used to find total of amount
```

	fraudAmount
•	24989780696.719975

Determine the locations with the highest number of fraudulent transactions.

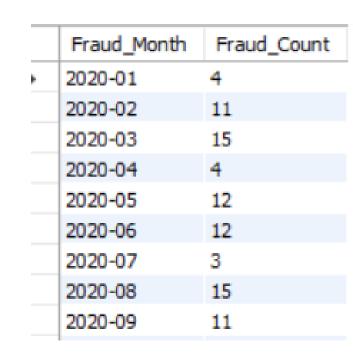
	location	HighFraudAmount
•	Paris	6364637354
	New York	4763636438
	Sydney	2748754789
	London	2647567475
	Mumbai	218374642.1

```
select location,max(transaction_amount) as HighFraudAmount
from fraud_dection
where is_fraud = 1
group by location
order by HighFraudAmount desc;
```

Max is use to find the maximum ammount from the transection amount column and we group it by location

Analyze the trend of fraudulent transactions per month.

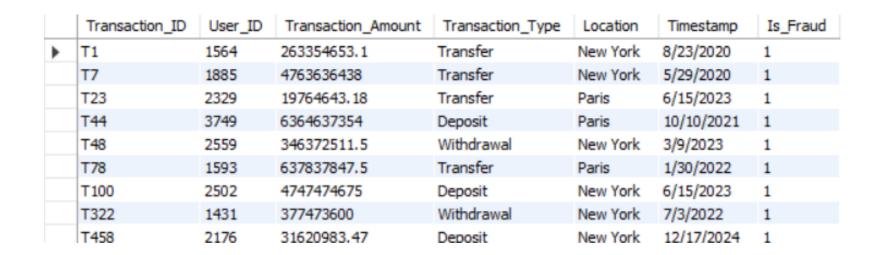
```
select DATE_FORMAT(STR_TO_DATE(Timestamp, '%m/%d/%Y'), '%Y-%m') AS Fraud_Month, COUNT(*) AS Fraud_Count
FROM fraud_dection
WHERE Is_Fraud = 1
GROUP BY Fraud_Month
ORDER BY Fraud_Month;
```



Date format helps here to format the time stamp in year-month

Retrieve high-value transactions (over 1,000,000) in New York and Paris.

```
select *
from fraud_dection
where transaction_amount > 10000000 and location in ("Paris", "New York");
```



Used "in" to find transactions from Paris and New York

Identify users with the highest number of transactions and their total transaction amounts.

```
select user_id,count(*) as total_transaction , sum(transaction_amount) as total_amount
from fraud_dection
group by user_id
order by total transaction desc;
```

	user_id	total_transaction	total_amount
•	1692	3	1008943.47
	3904	3	16276.04
	3160	3	757645971.64
	1618	3	17559.78
	1586	3	25247.02
	3680	3	17284.66
	2838	2	10158.68
	4881	2	5715.41
	1590	2	13454.810000000001

Used order by order it and desc to order it in deciding order

Determine which transaction types are most associated with fraudulent activities.

	transaction_type	fraud_type
•	Transfer	203
	Deposit	174
	Withdrawal	152

```
select transaction_type, count(transaction_type) as fraud_type
from fraud_dection
where is_fraud = 1
group by transaction_type
order by fraud_type desc;
```

Identify the periods with the highest occurrence of fraud.

	Fraud_Hour	Fraud_Count
•	03	60
	08	50
	06	50
	12	48
	09	46
	01	43
	02	42
	11	41
	10	40

Identify the periods with the highest occurrence of fraud.

```
select transaction_amount,transaction_type,
dense_rank() over (partition by transaction_type order by transaction_amount desc) as Highrank
from fraud_dection
limit 10;
```

Used dense_rank to rank the amounts dense_rank helps to rank without skipping if any amount repeats more then once

	transaction_amount	transaction_type	Highrank
>	6364637354	Deposit	1
	4747474675	Deposit	2
	757635374	Deposit	3
	328747474.3	Deposit	4
	37087638.47	Deposit	5
	35654613.1	Deposit	6
	31620983.47	Deposit	7
	4656679.34	Deposit	8
	3000087.6	Deposit	9
	997800	Deposit	10

Disclaimer

The dataset is just for practice and not a real dataset

I used the window function, Str_to_date, Aggregate functions, order by

In this dataset, I am dictating fraud in XYZ bank and the pattern of fraud