1. Columns and their datatypes:

- step (int, null)
- type (nvarchar(50), null)
- amount (float, null)
- nameOrig (nvarchar(50), null)
- oldbalanceOrg (float, null)
- newbalanceOrig (float, null)
- nameDest (nvarchar(50), null)
- oldbalanceDest (float, null)
- newbalanceDest (float, null)
- isFraud (int, null)
- isFlaggedFraud (int, null)

2. What type of transactions and total amount per type

```
select
    type,
    count (*) as amount
from Fraud
group by
    type
order by
    amount desc;
```

	type	amount
1	CASH_OUT	2237500
2	PAYMENT	2151495
3	CASH_IN	1399284
4	TRANSFER	532909
5	DEBIT	41432

3. Which types of transactions are fraudulent?

We find that of the five types of transactions, fraud occurs only in two of them 'TRANSFER' where money is sent to a customer / fraudster and 'CASH_OUT' where money is sent to a merchant who pays the customer / fraudster in cash.

These observations appear, at first, to bear out the description provided on Kaggle for the modus operandi of fraudulent transactions in this dataset, namely, fraud is committed by first transferring out funds to another account which subsequently cashes it out.

There are 2 flags which stand out to me and it's interesting to look onto: **isFraud** and **isFlaggedFraud** column. From the hypothesis, *isFraud* is the indicator which indicates the actual fraud transactions whereas *isFlaggedFraud* is what the system prevents the transaction due to some thresholds being triggered.

```
select
       count(*) as Transfer
```

```
from
	fraud
where
	  type = 'TRANSFER' and isFraud = 1

select
	count(*) as Cash_Out
from
	fraud
where
	  type = 'CASH_OUT' and isFraud = 1
```

The types of fraudulent transactions are ['TRANSFER', 'CASH_OUT']

The number of fraudulent TRANSFERs = 4097

The number of fraudulent CASH_OUTs = 4116

4. **What determines whether the feature *isFlaggedFraud* gets set or not?**

It turns out that the origin of *isFlaggedFraud* is unclear, contrasting with the description provided. The 16 entries (out of 6 million) where the *isFlaggedFraud* feature is set do not seem to correlate with any explanatory variable. The data is described as *isFlaggedFraud* being set when an attempt is made to 'TRANSFER' an 'amount' greater than 200,000. In fact, as shown below, *isFlaggedFraud* can remain not set despite this condition being met.

```
select
	count(*) as Flagged_Fraud
from
	Fraud
where

	isFlaggedFraud = 1

select
	type,
	amount,
	isFraud,
	isFlaggedFraud
from
	Fraud
where

isFlaggedFraud = 1

select
	type,
	amount,
	isFraud,
	isFlaggedFraud
from
	Fraud
where
	amount>200000
order by
	amount
desc
```

```sql
select
	min(amount) as minimum
from
	Fraud
where
	isFlaggedFraud = 1


select
	max(amount) as maximum
from
	Fraud
where
	isFlaggedFraud = 0
```

	Flagged_Fraud
1	16

	type	amount	isFraud	isFlaggedFraud
1	TRANSFER	536624.4375	1	1
2	TRANSFER	3576297	1	1
3	TRANSFER	353874.21875	1	1
4	TRANSFER	2542664.25	1	1
5	TRANSFER	3441041.5	1	1
6	TRANSFER	NULL	1	1
7	TRANSFER	7316255	1	1
8	TRANSFER	5674548	1	1
9	TRANSFER	4892193	1	1
10	TRANSFER	NULL	1	1
11	TRANSFER	9585040	1	1
12	TRANSFER	4953893	1	1
13	TRANSFER	1343002.125	1	1
14	TRANSFER	NULL	1	1
15	TRANSFER	399045.09375	1	1
16	TRANSFER	3171085.5	1	1

	type	amount	isFraud	isFlaggedFraud
1	TRANSFER	9998489	0	0
2	TRANSFER	9998070	0	0
3	TRANSFER	9997269	0	0
4	TRANSFER	9996887	1	0
5	CASH_OUT	9996887	1	0
6	TRANSFER	9996370	0	0
7	TRANSFER	9988600	0	0
8	TRANSFER	9982610	0	0
9	TRANSFER	9977761	1	0
10	CASH_OUT	9977761	1	0
11	TRANSFER	9977001	0	0
12	TRANSFER	9973371	0	0
13	TRANSFER	9972876	0	0
14	TRANSFER	9971121	0	0
15	TRANSFER	9969852	0	0
16	TRANSFER	9968663	0	0

	minimum
1	353874.21875

	maximum
1	9998489

The type of transactions in which isFlaggedFraud is set: ['TRANSFER']

Min amount transacted when isFlaggedFraud is set= 353874.22

Max amount transacted in a TRANSFER where isFlaggedFraud is not set= 9998489

Can *oldBalanceDest* and *newBalanceDest* determine *isFlaggedFraud* being set? The o ld is identical to the new balance in the origin and destination accounts, for every TRA NSFER where *isFlaggedFraud* is set. This is presumably because the transaction is ha lted. Interestingly, *oldBalanceDest* = 0 in every such transaction. However, as shown b elow, since *isFlaggedFraud* can remain not set in TRANSFERS where *oldBalanceDes t* and *newBalanceDest* can both be 0, these conditions do not determine the state of *is FlaggedFraud*.

```
select
       count(*) No_of_Transfers
from fraud
where
       isFlaggedFraud = 0 and
       oldBalanceDest = 0 and
       newBalanceDest = 0 and
       type = 'TRANSFER'
```

	No_of_Transfers
1	4158

isFlaggedFraud being set cannot be thresholded on *oldBalanceOrig* since the corresponding ran ge of values overlaps with that for TRANSFERs where *isFlaggedFraud* is not set (see below).

```
select
       min(oldbalanceOrg) as minimum,
       max(oldbalanceOrg) as maximum
from Fraud
where
        isFlaggedFraud = 1 and type = 'TRANSFER'


select
       min(oldbalanceOrg) as minimum,
       max(oldbalanceOrg) as maximum
from Fraud
where
       isFlaggedFraud = 0 and type = 'TRANSFER' and
        oldbalanceOrg = newbalanceOrig
```

	minimum	maximum
1	353874.21875	5674548

	minimum	maximum
1	0	575667.5625

Min, Max of oldbalanceOrg for isFlaggedFraud = 1 TRANSFERs: [353874.218, 5674548]

Min, Max of oldbalanceOrg for isFlaggedFraud = 0 TRANSFERs where oldbalanceOrg = newBalanceOrig: [0.0, 575667.5625]

Can *isFlaggedFraud* be set based on seeing a customer transacting more than once? Note that duplicate customer names don't exist within transactions where *isFlaggedFraud* is set, but duplicate customer names exist within transactions where *isFlaggedFraud* is not set. It turns out that originators of transactions that have *isFlaggedFraud* set have transacted only once. Very few destination accounts of transactions that have *isFlaggedFraud* set have transacted more than once.

```
Select
        *,
        RANK() over(order by nameDest) as dr
from
        Fraud
where
        isFlaggedFraud = 1
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	dr
1	671	TRANSFER	3441041.5	C917414431	3441041.5	3441041.5	C1082139865	0	0	1	1	1
2	279	TRANSFER	536624.4375	C1035541766	536624.4375	536624.4375	C1100697970	0	0	1	1	2
3	425	TRANSFER	9585040	C452586515	NULL	NULL	C1109166882	0	0	1	1	3
4	250	TRANSFER	1343002.125	C1100582606	1343002.125	1343002.125	C1147517658	0	0	1	1	4
5	702	TRANSFER	3171085.5	C1892216157	3171085.5	3171085.5	C1308068787	0	0	1	1	5
6	741	TRANSFER	5674548	C992223106	5674548	5674548	C1366804249	0	0	1	1	6
7	425	TRANSFER	NULL	C689608084	NULL	NULL	C1392803603	0	0	1	1	7
8	730	TRANSFER	NULL	C2140038573	NULL	NULL	C1395467927	0	0	1	1	8
9	586	TRANSFER	353874.21875	C1684585475	353874.21875	353874.21875	C1770418982	0	0	1	1	9
10	646	TRANSFER	NULL	C19004745	NULL	NULL	C1806199534	0	0	1	1	10
11	730	TRANSFER	7316255	C1869569059	NULL	NULL	C1861208726	0	0	1	1	11
12	646	TRANSFER	399045.09375	C724693370	NULL	NULL	C1909486199	0	0	1	1	12
13	554	TRANSFER	3576297	C193696150	3576297	3576297	C484597480	0	0	1	1	13
14	212	TRANSFER	4953893	C728984460	4953893	4953893	C639921569	0	0	1	1	14
15	617	TRANSFER	2542664.25	C786455622	2542664.25	2542664.25	C661958277	0	0	1	1	15
16	387	TRANSFER	4892193	C908544136	4892193	4892193	C891140444	0	0	1	1	16

```
Select
       *,
       RANK() over(order by nameDest) as dr
from
       Fraud
where
         isFlaggedFraud = 0
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	dr
1	370	TRANSFER	1331743	C1539355936	11088	0	C1000004082	228252.328125	1559995.25	0	0	1
2	352	CASH_IN	156985.3125	C1180747031	36186	193171.3125	C1000004082	0	0	0	0	1
3	354	CASH_OUT	228252.328125	C1978911345	953	0	C1000004082	0	228252.328125	0	0	1
4	396	CASH_IN	23297.189453125	C984964842	1030993.3125	1054290.5	C1000004082	1767010.25	1743713	0	0	1
5	374	CASH_OUT	363030.75	C1680720313	19486	0	C1000004082	1559995.25	1923026	0	0	1
6	379	CASH_IN	156015.828125	C1185840905	55451	211466.828125	C1000004082	1923026	1767010.25	0	0	1
7	131	CASH_OUT	269626.34375	C1456984251	62359.6484375	0	C1000004940	0	269626.34375	0	0	7
8	132	CASH_IN	368907.15625	C272149483	18824	387731.15625	C1000004940	269626.34375	0	0	0	7
9	212	CASH_OUT	61453.62109375	C1014009702	31984	0	C1000004940	362611.5625	424065.1875	0	0	7
10	161	CASH_OUT	181650.640625	C1953934428	29979	0	C1000004940	0	181650.640625	0	0	7
11	180	CASH_OUT	4445.02978515625	C1794226744	7032	2586.96997070313	C1000004940	181650.640625	186095.671875	0	0	7
12	207	CASH_OUT	176515.90625	C1621024259	0	0	C1000004940	186095.671875	362611.5625	0	0	7
13	355	CASH_IN	109120.4296875	C1191345963	2438982.25	2548102.75	C1000004940	832880.1875	723759.75	0	0	7
14	299	CASH_OUT	246485.6875	C553436408	0	0	C1000004940	424065.1875	670550.875	0	0	7
15	306	TRANSFER	228479.109375	C1075962941	0	0	C1000004940	604401.0625	832880.1875	0	0	7
16	303	CASH_OUT	111659.0625	C1816575322	0	0	C1000004940	670550.875	604401.0625	0	0	7
17	303	CASH_IN	177808.859375	C32214722	NULL	NULL	C1000004940	782209.9375	604401.0625	0	0	7
18	371	CASH_OUT	41067.19140625	C1497059448	42168	1100.81005859375	C1000004940	723759.75	764826.9375	0	0	7
19	377	TRANSFER	556785	C1861381106	169	0	C1000004940	764826.9375	1321612	0	0	7
20	202	CASH_IN	177276.359375	C628894690	393999.375	571275.75	C1000013769	243063.59375	65787.2265625	0	0	20

Have originators of transactions flagged as fraud transacted more than once? False

Have destinations for transactions flagged as fraud initiated other transactions? False

How many destination accounts of transactions flagged as fraud have been destination accounts more than once?: 0

It can be easily seen that transactions with *isFlaggedFraud* set occur at all values of *step*, similar to the complementary set of transactions. Thus *isFlaggedFraud* does not correlate with *step* either and is therefore seemingly unrelated to any explanatory variable or feature in the data

Conclusion: Although *isFraud* is always set when *isFlaggedFraud* is set, since *isFlaggedFraud* is set just 16 times in a seemingly meaningless way, we can treat this feature as insignificant and discard it in the dataset without loosing information.

5. **Are expected merchant accounts accordingly labelled?**

It was stated that CASH_IN involves being paid by a merchant (whose name is prefixed by 'M'). However, as shown below, the present data does not have merchants making CASH_IN transact ions to customers.

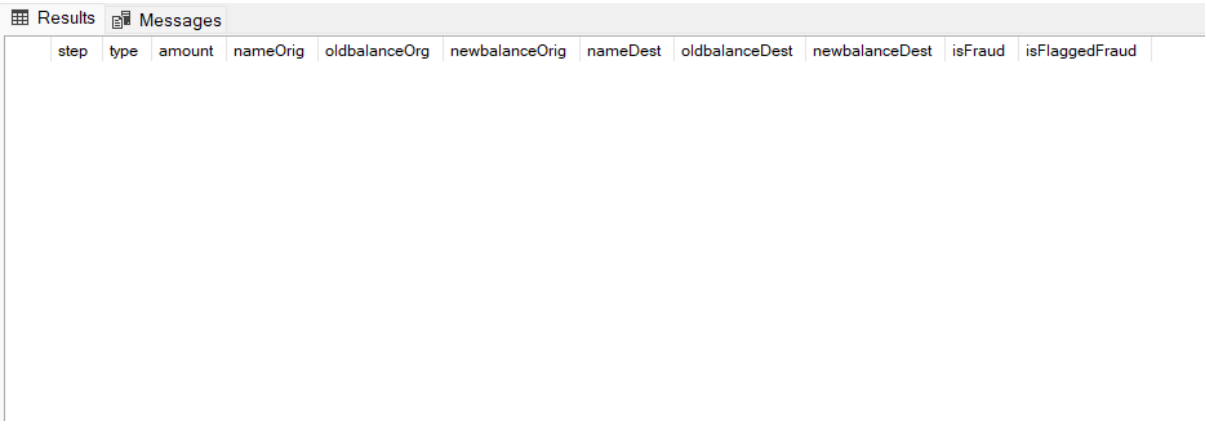
```
select
	*
from
	Fraud
where
	type = 'CASH_IN' and nameOrig like '%M%'
```

Results Messages

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud

Similarly, it was stated that CASH_OUT involves paying a merchant. However, for CASH_OUT tr ansactions there are no merchants among the destination accounts.

```
select
	*
from
	Fraud
where
	  type = 'CASH_OUT' and nameDest like '%M%'
```



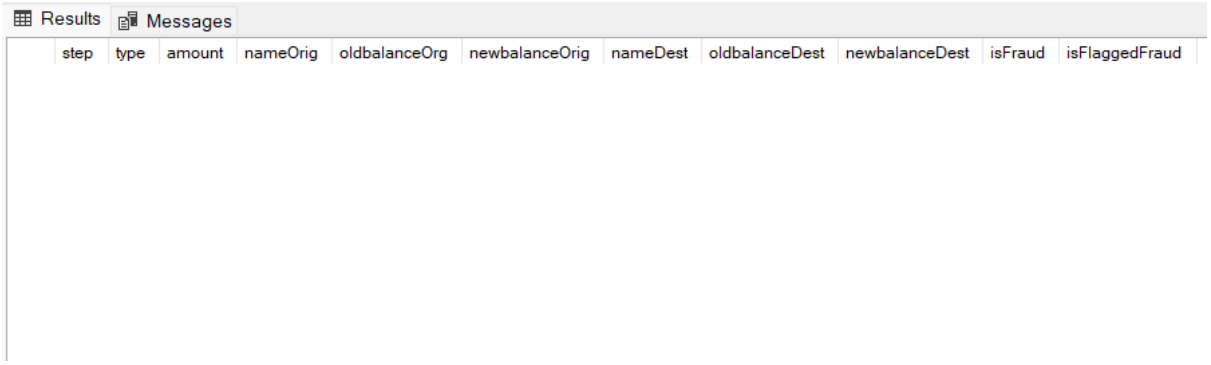
In fact, there are no merchants among any originator accounts. Merchants are only present in de stination accounts for all PAYMENTS.

Conclusion: Among the account labels *nameOrig* and *nameDest*, for all transactions, the mercha nt prefix of 'M' occurs in an unexpected way.

6. Are there account labels common to fraudulent TRANSFERs and CASH_OUTs?

From the data description, the modus operandi for committing fraud involves first making a TRAN SFER to a (fraudulent) account which in turn conducts a CASH_OUT. CASH_OUT involves trans acting with a merchant who pays out cash. Thus, within this two-step process, the fraudulent acc ount would be both, the destination in a TRANSFER and the originator in a CASH_OUT. Howeve r, the data shows below that there are no such common accounts among fraudulent transactions. Thus, the data is not imprinted with the expected modus-operandi.

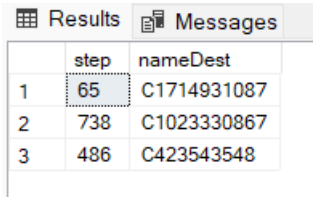
```
select
	*
from
	Fraud
where
	isFraud = 1 and
	nameDest in (select nameOrig from Fraud where isFraud = 1)
```



Within fraudulent transactions, are there destinations for TRANSFERS that are also originators for CASH _OUTs? False

Could destination accounts for fraudulent TRANSFERs originate CASHOUTs that are not detect ed and are labeled as genuine? It turns out there are 3 such accounts.

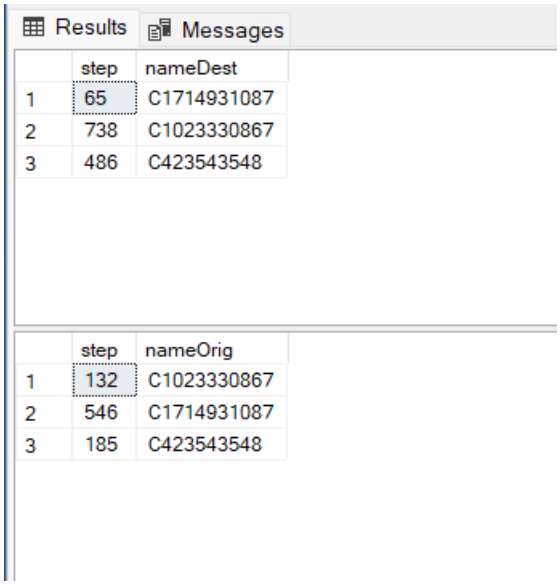
```
select
	step,nameDest
from
	Fraud
where
	 type = 'TRANSFER' and isFraud = 1 and
	  nameDest in (select nameOrig from Fraud where type = 'CASH_OUT' and isFraud =
0)
```



	step	nameDest
1	65	C1714931087
2	738	C1023330867
3	486	C423543548

```
select
	step,nameDest
from
	Fraud
where
	 type = 'TRANSFER' and isFraud = 1 and
	nameDest in (select nameOrig from Fraud where type = 'CASH_OUT' and isFraud =
0)


select
	step,nameOrig
from
	Fraud
where
	 type = 'CASH_OUT' and isFraud = 0 and
	nameOrig in ('C1714931087',
'C1023330867',
'C423543548')
```



Results Messages

	step	nameDest
1	65	C1714931087
2	738	C1023330867
3	486	C423543548

	step	nameOrig
1	132	C1023330867
2	546	C1714931087
3	185	C423543548

Fraudulent TRANSFER to C423543548 occured at step = 486 whereas genuine CASH_OUT from this acc ount occured earlier at step = 185