

USE retail\_banking\_analytics;

-- MODULE 3

-- Fraud & Transaction Risk Analysis

-- Task 1

-- Identify unusually large withdrawals relative to historical customer behavior

-- Computing average transaction size per account

```
SELECT
    account_id,
    AVG (ABS (amount)) AS avg_txn_amount -- withdrawal may be negative
FROM transactions
GROUP BY account_id;
```

account_id	avg_txn_amount
103762	33417.065000
101226	93731.484000
111034	34302.303333
113887	37689.863333
109697	44276.300000
107197	26597.970000
112727	73117.763333
106391	69561.330000
111732	28839.962500

-- Flag unusually large withdrawals

-- i.e. A transaction > 3 x account's average transaction size

```
SELECT
    t.transaction_id,
    t.account_id,
    t.transaction_date,
    t.amount,
    a.avg_txn_amount
FROM transactions t
JOIN (
    SELECT
```

```

        account_id,
        AVG (ABS (amount)) AS avg_txn_amount
    FROM transactions
    GROUP BY account_id
) a
    ON t. account_id = a. account_id
WHERE t. amount < 0
    AND ABS (t. amount) > 3 * a. avg_txn_amount;

```

## -- Task 2

### -- Fraud distribution by transaction channel

```

SELECT
    t. channel,
    COUNT (f. transaction_id) AS confirmed_fraud_count
FROM transactions t
LEFT JOIN fraud_flags f
    ON t. transaction_id = f. transaction_id
    AND f. status = 'Confirmed'
GROUP BY t. channel
ORDER BY confirmed_fraud_count DESC;

```

channel	confirmed_fraud_count
Mobile	183
POS	148
ATM	127
Branch	40

## -- Task 3

### -- Fraud rate by channel (CORRECT denominator logic)

```

SELECT
    t. channel,
    COUNT (*) AS total_transactions,
    SUM (

```

```

CASE
    WHEN f. status = 'Confirmed' THEN 1
    ELSE 0
END
) AS fraud_transactions,
ROUND (
    SUM (
        CASE
            WHEN f. status = 'Confirmed' THEN 1
            ELSE 0
        END
    ) / COUNT (*) * 100,
    2
) AS fraud_rate_pct
FROM transactions t
LEFT JOIN fraud_flags f
    ON t. transaction_id = f. transaction_id
GROUP BY t. channel
ORDER BY fraud_rate_pct DESC;

```

channel	total_transactions	fraud_transactions	fraud_rate_pct
Mobile	17565	183	1.04
ATM	12581	127	1.01
POS	14936	148	0.99
Branch	4918	40	0.81

**-- Task 4**  
**-- Accounts with repeated confirmed fraud**

```

SELECT
    t. account_id,
    COUNT (*) AS fraud_count
FROM transactions t
JOIN fraud_flags f
    ON t. transaction_id = f. transaction_id

```

```
WHERE f. status = 'Confirmed'  
GROUP BY t. account_id  
HAVING COUNT (*) > 1  
ORDER BY fraud_count DESC;
```

account_id	fraud_count
112806	2
107379	2
106910	2
101554	2
111665	2
106469	2
102497	2
108453	2
100198	2
110522	2
112781	2
105940	2