**SQL JOINING QUERIES**

**Created cust\_info table and changed the data type.**

CREATE TABLE IF NOT EXISTS cust\_info AS

SELECT CONVERT\_TO\_INTEGER(CustId, 1, 1, 0) AS CustId, CustName, CONVERT\_TO\_INTEGER(Age, 1, 1, 0) AS Age, Gender, Address

FROM sourceFile."first-case"."cust\_info.csv" AS "cust\_info.csv";

**Created transaction\_info table and changed the data type.**

CREATE TABLE IF NOT EXISTS transaction\_info AS

SELECT TO\_TIME("transactions\_20240116.csv"."Time", 'HH24:MI:SS', 1) AS "Time", TO\_DATE("transactions\_20240116.csv"."Date", 'DD/MM/YY', 1) AS "Date", CONVERT\_TO\_INTEGER(Sender\_account, 1, 1, 0) AS Sender\_account, CONVERT\_TO\_INTEGER(Receiver\_account, 1, 1, 0) AS Receiver\_account, CONVERT\_TO\_FLOAT(Amount, 1, 1, 0) AS Amount, Payment\_currency, Received\_currency, Sender\_bank\_location, Receiver\_bank\_location, Payment\_type, CONVERT\_TO\_INTEGER(Is\_laundering, 1, 1, 0) AS Is\_laundering, Laundering\_type

FROM sourceFile."first-case"."transactions\_20240116.csv" AS "transactions\_20240116.csv";

**1.Top 10 origin account number being used for fraud transactions**

CREATE TABLE IF NOT EXISTS Fraud\_Transaction AS

SELECT Sender\_account, COUNT(\*) AS Fraud\_Transaction\_Count

FROM transaction\_info

WHERE Is\_laundering = 1

GROUP BY Sender\_account

ORDER BY Fraud\_Transaction\_Count DESC

LIMIT 10;

SELECT \* FROM Fraud\_Transaction;

**2. Top 10 destination account number being used for fraud transactions**

CREATE TABLE IF NOT EXISTS Fraud\_Transaction\_dest AS

SELECT Receiver\_account, COUNT(\*) AS Fraud\_Transaction\_Count

FROM transaction\_info

WHERE Is\_laundering = 1

GROUP BY Receiver\_account

ORDER BY Fraud\_Transaction\_Count DESC

LIMIT 10;

**3. Top 3 most used currency**

CREATE TABLE IF NOT EXISTS Currency\_Usage AS

SELECT Payment\_currency, COUNT(\*) AS Currency\_Usage\_Count

FROM transaction\_info

GROUP BY Payment\_currency

ORDER BY Currency\_Usage\_Count DESC

LIMIT 3;

SELECT \* FROM Currency\_Usage;

**4. Top 3 sender and receiver bank locations**

CREATE TABLE IF NOT EXISTS Transaction\_Count\_Table AS

SELECT Bank\_Location, COUNT(\*) AS Transaction\_Count

FROM (

SELECT Sender\_bank\_location AS Bank\_Location FROM transaction\_info

UNION ALL

SELECT Receiver\_bank\_location AS Bank\_Location FROM transaction\_info

) AS combined\_locations

GROUP BY Bank\_Location

ORDER BY Transaction\_Count DESC

LIMIT 3;

SELECT \* FROM Transaction\_Count\_Table;

**5. Age group where most of the fraud transactions take place. define age group like the following 15-20, 20-25, 25-30 etc**

CREATE TABLE IF NOT EXISTS Age\_Group\_Fraud\_Transaction\_Count\_Table AS

SELECT

CASE

WHEN Age BETWEEN 15 AND 20 THEN '15-20'

WHEN Age BETWEEN 21 AND 25 THEN '21-25'

WHEN Age BETWEEN 26 AND 30 THEN '26-30'

WHEN Age BETWEEN 31 AND 35 THEN '31-35'

WHEN Age BETWEEN 36 AND 40 THEN '36-40'

WHEN Age BETWEEN 41 AND 45 THEN '41-45'

WHEN Age BETWEEN 46 AND 50 THEN '46-50'

ELSE '51+'

END AS Age\_Group\_Criteria,

COUNT(\*) AS Age\_Group\_Fraud\_Transaction\_Count

FROM transaction\_info

JOIN cust\_info ON transaction\_info.Sender\_account = cust\_info.CustId

WHERE Is\_laundering = 1

GROUP BY Age\_Group\_Criteria

ORDER BY Age\_Group\_Fraud\_Transaction\_Count DESC

LIMIT 1;

**6. top 3 most used launder type for money laundering. top 3 launder type for non-regular transactions**

CREATE TABLE IF NOT EXISTS Laundering\_Details\_Table AS

WITH Fraud\_Laundering AS (

SELECT

Laundering\_type,

COUNT(\*) AS Transaction\_Count,

'Fraud' AS Transaction\_Type

FROM transaction\_info

WHERE Is\_laundering = 1

GROUP BY Laundering\_type

ORDER BY Transaction\_Count DESC

LIMIT 3

),

Non\_Fraud\_Laundering AS (

SELECT

Laundering\_type,

COUNT(\*) AS Transaction\_Count,

'Non-Fraud' AS Transaction\_Type

FROM transaction\_info

WHERE Is\_laundering = 0

GROUP BY Laundering\_type

ORDER BY Transaction\_Count DESC

LIMIT 3

)

SELECT \* FROM Fraud\_Laundering

UNION ALL

SELECT \* FROM Non\_Fraud\_Laundering;

SELECT \* FROM Laundering\_Details\_Table;

**7. Countries who have initiated most cross-border transaction**

CREATE TABLE IF NOT EXISTS Cross\_Border\_Transaction\_Table AS

SELECT Sender\_bank\_location, COUNT(\*) AS Transaction\_Count

FROM transaction\_info

WHERE Sender\_bank\_location != Receiver\_bank\_location

GROUP BY Sender\_bank\_location

ORDER BY Transaction\_Count DESC

LIMIT 10;

SELECT \* FROM Cross\_Border\_Transaction\_Table;