--Combine dataset

drop view if exists dt1;

Create VIEW dt1 as

SELECT

cm.Client\_ID,

cm.Client\_Segment,

kyc.\*,

ab\_202411.balance AS balance\_202411,

ab\_202412.balance AS balance\_202412,

ab\_202501.balance AS balance\_202501,

ab\_202502.balance AS balance\_202502,

ab\_202503.balance AS balance\_202503,

ab\_202504.balance AS balance\_202504

FROM client\_mapping cm

INNER JOIN kyc

ON cm.Account\_ID = kyc.Account\_ID

RIGHT JOIN account\_balance\_202411 ab\_202411

ON cm.account\_id = ab\_202411.account\_id

RIGHT JOIN account\_balance\_202412 ab\_202412

ON cm.account\_id = ab\_202412.account\_id

RIGHT JOIN account\_balance\_202501 ab\_202501

ON cm.account\_id = ab\_202501.account\_id

RIGHT JOIN account\_balance\_202502 ab\_202502

ON cm.account\_id = ab\_202502.account\_id

RIGHT JOIN account\_balance\_202503 ab\_202503

ON cm.account\_id = ab\_202503.account\_id

RIGHT JOIN account\_balance\_202504 ab\_202504

ON cm.account\_id = ab\_202504.account\_id

ORDER BY cm.Client\_ID, cm.account\_id;

--Create kpi’s

CREATE view kpi as

WITH monthly\_sums AS (

SELECT

Client\_ID,

SUM(COALESCE(balance\_202411, 0)) AS bal\_202411,

SUM(COALESCE(balance\_202412, 0)) AS bal\_202412,

SUM(COALESCE(balance\_202501, 0)) AS bal\_202501,

SUM(COALESCE(balance\_202502, 0)) AS bal\_202502,

SUM(COALESCE(balance\_202503, 0)) AS bal\_202503,

SUM(COALESCE(balance\_202504, 0)) AS bal\_202504

FROM dt1

GROUP BY Client\_ID

)

SELECT

d.Client\_ID,

MAX(d.Client\_Segment) AS Client\_Segment,

MAX(d.ClientLanguage) AS ClientLanguage,

MAX(d.BirthDate) AS BirthDate,

-- Null balance KPIs (all accounts have zero in recent N months)

CASE WHEN ms.bal\_202504 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_1m,

CASE WHEN ms.bal\_202504 + ms.bal\_202503 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_2m,

CASE WHEN ms.bal\_202504 + ms.bal\_202503 + ms.bal\_202502 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_3m,

CASE WHEN ms.bal\_202504 + ms.bal\_202503 + ms.bal\_202502 + ms.bal\_202501 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_4m,

CASE WHEN ms.bal\_202504 + ms.bal\_202503 + ms.bal\_202502 + ms.bal\_202501 + ms.bal\_202412 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_5m,

CASE WHEN ms.bal\_202504 + ms.bal\_202503 + ms.bal\_202502 + ms.bal\_202501 + ms.bal\_202412 + ms.bal\_202411 = 0 THEN 1 ELSE 0 END AS is\_null\_balance\_6m,

-- Total balances by month

ms.bal\_202411 AS total\_balance\_202411,

ms.bal\_202412 AS total\_balance\_202412,

ms.bal\_202501 AS total\_balance\_202501,

ms.bal\_202502 AS total\_balance\_202502,

ms.bal\_202503 AS total\_balance\_202503,

ms.bal\_202504 AS total\_balance\_202504,

-- Net change in total balance (latest - earliest)

ms.bal\_202504 - ms.bal\_202411 AS net\_change\_total\_balance,

-- Active account count

SUM(CASE WHEN COALESCE(d.balance\_202411, 0) > 0 THEN 1 ELSE 0 END) AS active\_accounts\_202411,

SUM(CASE WHEN COALESCE(d.balance\_202504, 0) > 0 THEN 1 ELSE 0 END) AS active\_accounts\_202504,

SUM(CASE WHEN COALESCE(d.balance\_202504, 0) > 0 THEN 1 ELSE 0 END) -

SUM(CASE WHEN COALESCE(d.balance\_202411, 0) > 0 THEN 1 ELSE 0 END) AS net\_change\_active\_accounts,

-- Churn flag (all accounts zero in last month)

CASE WHEN ms.bal\_202504 = 0 THEN 1 ELSE 0 END AS churn\_flag,

-- Closed accounts: balance >0 at start, zero at end

SUM(CASE WHEN COALESCE(d.balance\_202411, 0) > 0 AND COALESCE(d.balance\_202504, 0) = 0 THEN 1 ELSE 0 END) AS closed\_accounts\_count,

-- Max/min total monthly balances (across 6 months)

(

SELECT MAX(val) FROM (

SELECT ms.bal\_202411 AS val UNION ALL

SELECT ms.bal\_202412 UNION ALL

SELECT ms.bal\_202501 UNION ALL

SELECT ms.bal\_202502 UNION ALL

SELECT ms.bal\_202503 UNION ALL

SELECT ms.bal\_202504

)

) AS max\_balance\_any\_month,

(

SELECT MIN(val) FROM (

SELECT ms.bal\_202411 AS val UNION ALL

SELECT ms.bal\_202412 UNION ALL

SELECT ms.bal\_202501 UNION ALL

SELECT ms.bal\_202502 UNION ALL

SELECT ms.bal\_202503 UNION ALL

SELECT ms.bal\_202504

)

) AS min\_balance\_any\_month,

-- Average % change in monthly balance (client-level, all accounts summed)

(

(CASE WHEN ms.bal\_202411 != 0 THEN (ms.bal\_202412 - ms.bal\_202411) \* 1.0/ms.bal\_202411 ELSE NULL END) +

(CASE WHEN ms.bal\_202412 != 0 THEN (ms.bal\_202501 - ms.bal\_202412) \* 1.0/ms.bal\_202412 ELSE NULL END) +

(CASE WHEN ms.bal\_202501 != 0 THEN (ms.bal\_202502 - ms.bal\_202501) \* 1.0/ms.bal\_202501 ELSE NULL END) +

(CASE WHEN ms.bal\_202502 != 0 THEN (ms.bal\_202503 - ms.bal\_202502) \* 1.0/ms.bal\_202502 ELSE NULL END) +

(CASE WHEN ms.bal\_202503 != 0 THEN (ms.bal\_202504 - ms.bal\_202503) \* 1.0/ms.bal\_202503 ELSE NULL END)

) / 5.0 \* 100 AS avg\_pct\_change\_over\_periods

FROM dt1 d

JOIN monthly\_sums ms ON d.Client\_ID = ms.Client\_ID

GROUP BY d.Client\_ID;

--Creating more KPIs

WITH long\_balances AS (

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202411' AS Month, balance\_202411 AS Balance FROM dt1

UNION ALL

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202412' AS Month, balance\_202412 AS Balance FROM dt1

UNION ALL

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202501' AS Month, balance\_202501 AS Balance FROM dt1

UNION ALL

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202502' AS Month, balance\_202502 AS Balance FROM dt1

UNION ALL

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202503' AS Month, balance\_202503 AS Balance FROM dt1

UNION ALL

SELECT Client\_ID, Account\_ID, Client\_Segment, ClientLanguage, BirthDate, '202504' AS Month, balance\_202504 AS Balance FROM dt1

),

activity AS (

SELECT

Client\_ID,

Account\_ID,

Client\_Segment,

ClientLanguage,

BirthDate,

MIN(CASE WHEN Balance IS NOT NULL THEN Month END) AS first\_active\_month,

MAX(CASE WHEN Balance IS NOT NULL THEN Month END) AS last\_active\_month,

COUNT(CASE WHEN Balance IS NOT NULL THEN 1 END) AS months\_active,

SUM(CASE WHEN Month = '202504' AND Balance IS NOT NULL THEN 1 ELSE 0 END) AS active\_in\_last\_month

FROM long\_balances

GROUP BY Account\_ID

),

before\_closure\_balance AS (

-- Get the balance in the month before the last active month for churned accounts

SELECT

lb.Account\_ID,

MAX(lb.Month) AS before\_closure\_month,

MAX(lb.Balance) AS balance\_before\_closure

FROM long\_balances lb

JOIN activity a ON lb.Account\_ID = a.Account\_ID

WHERE lb.Month < a.last\_active\_month

AND lb.Balance IS NOT NULL

AND a.months\_active < 6 -- Churned account

GROUP BY lb.Account\_ID

)

SELECT

a.Client\_ID,

a.Account\_ID,

a.Client\_Segment,

a.ClientLanguage,

a.BirthDate,

a.first\_active\_month,

a.last\_active\_month,

a.months\_active,

CASE WHEN a.months\_active < 6 THEN 1 ELSE 0 END AS churn\_flag,

CASE WHEN a.months\_active < 6 THEN a.last\_active\_month ELSE NULL END AS closure\_month,

bc.balance\_before\_closure,

-- First and last non-null balances for % change

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.first\_active\_month) AS first\_balance,

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.last\_active\_month) AS last\_balance,

CASE

WHEN (SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.first\_active\_month) IS NOT NULL AND

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.last\_active\_month) IS NOT NULL AND

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.first\_active\_month) != 0

THEN

((SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.last\_active\_month) -

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.first\_active\_month)) \* 1.0 /

(SELECT Balance FROM long\_balances WHERE Account\_ID = a.Account\_ID AND Month = a.first\_active\_month)

ELSE NULL

END AS pct\_change\_first\_to\_last

FROM activity a

LEFT JOIN before\_closure\_balance bc ON a.Account\_ID = bc.Account\_ID;