```
USE retail_banking_analytics;

-- MODULE 2
-- Customer Financial Health Assessment

-- To confirm the relationship chain
SELECT
COUNT (*) AS total_accounts,
COUNT (customer_id) AS totaL_customers
FROM accounts;
```

total_accounts	totaL_customers
15000	15000

```
-- Task 1
-- Customers who fully repaid loans but have low balances

-- Total balance per customer
SELECT customer_id,
SUM (balance) AS total_customer_balance
FROM accounts
GROUP BY customer_id
ORDER BY total_customer_balance Desc;
```

customer_id	total_customer_balance
677	10667580.19
750	8273698.79
9265	7812429.59
1671	7607343.09
1747	6255094.43
5533	6204753.07
2339	5941912.40
9422	5828220.71
840	5520988.35
1972	5434674.30

-- Total principal repaid per customer (Closed loans only)

```
SELECT customer_id,
SUM (principal_amount) AS total_principal_repaid
FROM loans
WHERE loan_status = "Closed"
GROUP BY customer_id;
```

customer_id	total_principal_repaid
2593	205770.38
3757	221389.02
3713	118156.76
2348	2368723.77
5955	195721.03
2689	342294.87
655	99846.86
3111	432419.54
6203	509119.04
732	178547.91

-- Combine and flag customers

```
SELECT
        l.customer_id,
   SUM (l. principal_amount) AS total_principal_repaid,
   SUM (a. balance) AS total_customer_balance
FROM loans l
JOIN accounts a
        ON l. customer_id = a. customer_id
WHERE l. loan_status = "Closed"
GROUP BY l. customer_id
HAVING SUM (a. balance) < SUM (l. principal_amount);
```

customer_id	total_principal_repaid	total_customer_balance
8619	1216503.80	1118160.16
8670	4410436.88	1469578.48
5801	384925.70	256663.74
4526	1450980.68	1370704.97
1024	869542.12	356298.30
6077	8437088.25	1288088.77
5345	6336811.88	983311.05
5644	1172203.26	1158973.22
2192	362538.83	240455.23
3046	3363275.64	787550.33

-- Task 2

-- Customers who defaulted on loans but still maintain high account balances

-- Define High balance

```
SELECT AVG (balance) AS avg_balance
FROM accounts;
-- let's say High Balance = Above Average
```

avg_balance
297325.363021

-- Identify Customers

```
SELECT
        l.customer_id,
   SUM (l. principal_amount) AS total_defaulted_amount,
   SUM (a. balance) AS total_customer_balance
FROM loans l
JOIN accounts a
        ON l. customer_id = a. customer_id
WHERE l. loan_status = "Defaulted"
GROUP BY l. customer_id
HAVING SUM (a. balance) > (
        SELECT AVG (balance) FROM accounts
);
```

customer_id	total_defaulted_amount	total_customer_balance
2467	1924907.08	1141510.02
1741	1945345.44	2169240.10
1486	940825.82	836215.74
7245	1616522.22	356218.12
3046	1284597.54	787550.33
6252	117675.48	958098.37
3993	334900.30	1417795.17
8557	2840549.22	550814.41
6163	3200604.30	438712.04
8395	747569.02	1625538.89

Note: The tables were limited to 10 rows