

# Problem Statement

## Title

"Improving Customer Profitability, Risk Management, and Digital Engagement through Transaction-Level Analysis in Bank"

## Context:

Bank operates across multiple cities in Spain, offering diverse financial products and services such as savings accounts, credit cards, loans, and digital banking platforms. Despite having robust transactional infrastructure, management has limited visibility into **customer behaviour trends, revenue leakage from fees, inefficient product recommendations, and digital adoption patterns**.

The bank aims to **leverage its historical transaction data (2023–2025)** to uncover **analytical patterns** that will:

- Improve customer engagement and retention
- Optimize product cross-selling strategies
- Strengthen risk mitigation approaches
- Enhance the overall efficiency of channel operations

## Business Challenge

1. **Unidentified Overleveraged Customers:**  
Customers spending significantly more than their income may pose credit risks.
2. **Inefficient Product Recommendations:**  
Many recommended financial products do not match the actual usage or interest of customers.
3. **Fee Revenue Distribution Is Unclear:**  
The bank earns revenue from various fees (late payment, card, insurance), but branch-level and customer-segment-level contribution is unknown.
4. **Customer Loyalty and Retention Insights Are Lacking:**  
There is no visibility into how long customers stay active or whether they are disengaging over time.
5. **Digital Transformation Monitoring Is Needed:**  
The bank has been shifting toward digital channels but lacks analytics to verify behaviour changes and channel migration.
6. **Underused Cross-Sell Potential:**  
The management suspects that many customers hold only one or two product types when they could be using more.

## Objective

The goal of this analysis is to create a **data-driven foundation** for:

- Detecting **at-risk or high-value customers**
- Improving **personalized financial offerings**
- Understanding **fee structures** and their contribution
- Analysing **channel behaviour shifts** to guide future investments in mobile and online banking
- Quantifying **customer lifecycle and engagement** for retention strategies

## Dataset Explanation

This dataset contains a **comprehensive record of 20,000 banking transactions** conducted by customers across various cities in Spain between **2023 and 2025**. Each row represents a **unique banking transaction**, capturing critical aspects of customer behaviour, product interaction, transactional activity, and revenue-generating elements such as fees and penalties.

The dataset was imported into a **MySQL database** under the schema bank, with the working table named bank\_dataset.

## Purpose of Dataset

To enable in-depth **financial, behavioural, and operational analysis** of customers' banking activities. This includes:

- Understanding spending, saving, and borrowing habits
- Evaluating financial product usage and recommendations
- Tracking channel preferences (Branch, ATM, Mobile, Online)
- Identifying risk through late payments or high spending-to-income ratios
- Measuring branch-level and regional performance
- Analysing engagement through transaction frequency and tenure

Column Name	Type	Description
TransactionID	Integer	Unique identifier for each transaction (Primary Key)
CustomerID	Integer	Unique customer identifier
TransactionDate	DateTime	Timestamp of the transaction — used for time-based and cohort analysis
TransactionType	String	Nature of transaction: Deposit, Withdrawal, Card Payment, Loan Payment, etc.
Amount	Float	The monetary value of the transaction

<b>ProductCategory</b>	String	Main product type: Checking, Savings, Loan, Credit Card, Mortgage
<b>ProductSubcategory</b>	String	Further classification: Gold, Platinum, Student, etc.
<b>BranchCity</b>	String	City where the transaction occurred
<b>BranchLat, BranchLong</b>	Float	Latitude and longitude of the branch — used for geospatial insights
<b>Channel</b>	String	Mode of transaction: Online, Mobile, ATM, Branch
<b>Currency</b>	String	Currency of transaction (EUR or USD)
<b>CreditCardFees</b>	Float	Fees associated with credit card usage
<b>InsuranceFees</b>	Float	Insurance-related charges
<b>LatePaymentAmount</b>	Float	Penalties incurred due to late payments
<b>CustomerScore</b>	Integer	Internal scoring based on creditworthiness or engagement
<b>MonthlyIncome</b>	Float	Customer's reported monthly income
<b>CustomerSegment</b>	String	Segment classification: High, Middle, Low Income
<b>RecommendedOffer</b>	String	The bank's suggested financial product or benefit for the customer

## Key Analytical Potentials

- Temporal Analysis: Via TransactionDate for seasonality, trends, and churn indicators
- Behavioral Clustering: Based on transaction types, frequency, and amounts
- Fee Revenue Attribution: Using CreditCardFees, InsuranceFees, and LatePaymentAmount
- Channel Analysis: To measure the shift toward digital banking
- Geospatial Intelligence: Leveraging BranchCity, BranchLat, and BranchLong
- Risk Profiling: Using CustomerScore, income-spending patterns, and payment delays
- Product Strategy: Assessing the alignment of RecommendedOffer with actual customer behavior

## Queries to Find Answer

### Query 1 Customer Lifetime Value and Engagement Pattern




**Purpose:** To identify the most valuable customers based on the total money transacted and engagement level (transaction count).

**Business Use:** Helps the bank focus on top-tier clients for premium services, loyalty programs, and retention efforts.

use bank;

-- 1. Customer Lifetime Value and Engagement Pattern

```
SELECT
    CustomerID,
    SUM(Amount) AS TotalAmount,
    COUNT(*) AS TransactionCount,
    RANK() OVER (ORDER BY SUM(Amount)DESC) AS RankByValue
FROM
    bank_dataset
GROUP BY CustomerID;
```

Result Grid		 Filter Rows:	 Export:	 Wrap Cell
	CustomerID	TotalAmount	TransactionCount	RankByValue
▶	4239	69208.824807	7	1
	4644	58920.782546	6	2
	1948	56196.479506	7	3
	5887	55083.070601499996	7	4
	8749	54942.664605100006	6	5
	7028	53149.279540999996	6	6
	7998	52695.683559	6	7
	9125	50587.1635736	8	8
	5555	50169.825081999996	6	9
	9120	49787.777395	7	10
	7229	49286.973379	7	11
	1968	49264.850565	7	12
	6988	48911.6971661	7	13
	4086	48214.417954000004	5	14
	6622	48043.997447	7	15

### Query 2: Monthly Trends and Seasonality

**Purpose:** To analyse the volume and value of transactions by type across months.

**Business Use:** Understands seasonality in customer behaviour to plan campaigns (e.g., loan offers during high-spending months or card rewards in festive seasons).

-- 2. Monthly Trends and Seasonality

```
SELECT
    DATE_FORMAT(TransactionDate, '%Y-%m') AS Month,
```

```

TransactionType,
COUNT(*) AS Transactions,
SUM(Amount) AS TotalAmount
FROM
    bank_dataset
GROUP BY
    Month, TransactionType
ORDER BY
    Month, Transactions, TransactionType;

```

	Month	TransactionType	Transactions	TotalAmount
▶	2023-01	Deposit	108	555025.8082804997
	2023-01	Card Payment	114	553191.5964581901
	2023-01	Transfer	114	557906.7581149
	2023-01	Fee	124	680515.1958956303
	2023-01	Loan Payment	134	623279.9753740801
	2023-01	Withdrawal	138	649445.3098310799
	2023-02	Transfer	81	432019.2822542
	2023-02	Deposit	85	464474.50451750006
	2023-02	Card Payment	105	462725.21192389977
	2023-02	Loan Payment	110	577227.3912388999
	2023-02	Withdrawal	116	536188.54335252

Result 29 ×

### Query 3: Risk Flags – High Fees and Late Payments

**Purpose:** Identifies customers frequently incurring late payment fees and evaluates their income levels.

**Business Use:** Flags potential defaulters or financially stressed customers for risk assessment and credit policy tuning.

```

WITH RiskyLatePayers AS (
    SELECT
        CustomerID,
        COUNT(*) AS LateFeeCount,
        ROUND(SUM(LatePaymentAmount), 2) AS TotalLateFees,
        ROUND(AVG(MonthlyIncome), 2) AS AvgIncome
    FROM bank_dataset

```

```

WHERE LatePaymentAmount > 50
GROUP BY CustomerID
HAVING COUNT(*) > 2
)
SELECT *
FROM RiskyLatePayers
ORDER BY TotalLateFees DESC;

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:				
	CustomerID	LateFeeCount	TotalLateFees	AvgIncome
▶	4083	4	646	6098.24
	1845	4	637	6084.38
	4093	4	600	5917.44
	2301	3	531	5268.82
	2646	3	527	3224.75
	4718	4	524	4824.18
	5297	3	496	5680.08
	8529	3	492	6455.43
	6599	3	472	2698.52
	1224	3	470	5066.93
	4364	3	459	6219.91

Result 30 x

## Query 4: Channel Effectiveness and Customer Preference

**Purpose:** Breaks down transaction behavior across digital (Online, Mobile) vs physical (Branch, ATM) channels by customer segment.

**Business Use:** Helps in optimizing branch operations, improving digital UX, and reducing operational costs by promoting digital adoption.

-- 4. Channel Effectiveness and Customer Preference

```

SELECT
    CustomerSegment,
    Channel,
    COUNT(*) AS TotalTransactions,
    SUM(Amount) AS TotalAmount
FROM
    bank_dataset
GROUP BY
    CustomerSegment, Channel
order by

```

CustomerSegment;

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	CustomerSegment	Channel	TotalTransactions	TotalAmount			
▶	High Income Segment	ATM	1597	8120330.249858725			
	High Income Segment	Branch	1648	8269212.817237238			
	High Income Segment	Mobile	1757	8777437.257214678			
	High Income Segment	Online	1624	8173374.251013263			
	Low Income Segment	ATM	1130	5696710.050057711			
	Low Income Segment	Branch	1124	5577493.156720967			
	Low Income Segment	Mobile	1167	5658983.739139631			
	Low Income Segment	Online	1068	5215544.009302293			
	Middle Income Segment	ATM	2156	11288454.797126507			
	Middle Income Segment	Branch	2261	11456421.459625114			
	Middle Income Segment	Mobile	2295	11663384.302348405			
Result 31					×		

## Query 5: Recommendation System Effectiveness

**Purpose:** Evaluates whether recommended offers (like credit cards) are aligned with actual customer spending (card payments).

**Business Use:** Assesses the accuracy of marketing and recommendation algorithms to improve personalized offers.

- 5. Recommendation System Effectiveness

SELECT

RecommendedOffer,

COUNT(\*) AS OfferCount,

SUM(CASE WHEN TransactionType = 'Card Payment' THEN Amount ELSE 0 END) AS  
TotalCardPayments,

AVG(MonthlyIncome) AS AvgIncome

FROM

bank\_dataset



GROUP BY

RecommendedOffer

ORDER BY

TotalCardPayments DESC;



Result Grid				
Filter Rows:		Export:  Wrap Cell Content: 		
	RecommendedOffer	OfferCount	TotalCardPayments	AvgIncome
▶	Mid-tier Savings Booster	5220	3956926.19420777	5005.059390804601
	Financial Literacy Program Access	3540	3121478.06588426	2007.072983050854
	Premium Investment Services	3954	2983535.2448258908	8498.394984825502
	Exclusive Platinum Package	2672	2692981.9326866497	8524.439180389221
	Personal Loan Cashback Offer	1828	2059397.879587989	4960.796531728665
	Gold Card with Travel Benefits	1837	1507649.6397082694	4964.714594447477
	No-Fee Basic Account	949	358675.92984813004	2001.0762697576401

Result 32 x

## Query 6: City-Level Profitability and Geographical Patterns

**Purpose:** Maps where transaction volumes and fee revenue are highest geographically.

**Business Use:** Aids in regional performance assessment and helps decide where to expand or optimize branch operations.

-- 6. City-Level Profitability and Geographical Patterns

SELECT

BranchCity,

COUNT(\*) AS TotalTransactions,

SUM(Amount) AS TotalTransactionAmount,



SUM(CreditCardFees + InsuranceFees + LatePaymentAmount) AS TotalFees

FROM

bank\_dataset

GROUP BY BranchCity

ORDER BY TotalFees DESC;

Result Grid				
Filter Rows:		Export:  Wrap Cell Content: 		
	BranchCity	TotalTransactions	TotalTransactionAmount	TotalFees
▶	Barcelona	2564	13033624.542848308	85248.43000000001
	Malaga	2524	12917492.751719745	85090.15999999993
	Murcia	2564	12899902.763350684	85055.21999999996
	Madrid	2472	12181857.616868056	80297.15000000005
	Valencia	2432	12458382.31454798	76144.66999999995
	Bilbao	2455	12219141.947407855	75778.34000000005
	Zaragoza	2476	12623572.115856778	75612.81
	Seville	2513	12673128.297178242	73953.43999999999



## Query 7: Customer Churn Signals

**Purpose:** Finds mismatches between the product customers use most and what they're being offered.

**Business Use:** Misaligned recommendations can frustrate customers and lead to churn. Helps refine targeting to improve customer satisfaction and retention.

-- 7. Customer Churn Signals

```
WITH ProductUsage AS (  
    SELECT  
        CustomerID,  
        ProductCategory,  
        COUNT(*) AS UsageCount,  
        RANK() OVER (PARTITION BY CustomerID ORDER BY COUNT(*) DESC) AS rnk  
    FROM bank_dataset  
    GROUP BY CustomerID, ProductCategory  
,  
TopProduct AS (  
    SELECT CustomerID, ProductCategory AS MostUsedProduct  
    FROM ProductUsage  
    WHERE rnk = 1  
,  
OfferMapped AS (  
    SELECT  
        CustomerID,  
        RecommendedOffer,  
        CASE  
            WHEN RecommendedOffer LIKE '%Savings%' THEN 'Savings Account'  
            WHEN RecommendedOffer LIKE '%Loan%' THEN 'Loan'  
            WHEN RecommendedOffer LIKE '%Card%' THEN 'Credit Card'  
            WHEN RecommendedOffer LIKE '%Investment%' THEN 'Savings Account'  
            WHEN RecommendedOffer LIKE '%Account%' THEN 'Checking Account'  
            ELSE 'Other'  
        END AS MappedProduct  
    FROM bank_dataset  
,  
Mismatch AS (  
    SELECT  
        t.CustomerID,  
        t.MostUsedProduct,  
        o.MappedProduct,
```

```

        o.RecommendedOffer
FROM TopProduct t
JOIN OfferMapped o ON t.CustomerID = o.CustomerID
GROUP BY t.CustomerID, t.MostUsedProduct, o.MappedProduct, o.RecommendedOffer
HAVING t.MostUsedProduct != o.MappedProduct
)
SELECT * FROM Mismatch;

```

CustomerID	MostUsedProduct	MappedProduct	RecommendedOffer
8270	Checking Account	Savings Account	Mid-tier Savings Booster
1860	Mortgage	Other	Financial Literacy Program Access
6390	Loan	Savings Account	Premium Investment Services
6191	Mortgage	Other	Financial Literacy Program Access
6734	Checking Account	Savings Account	Mid-tier Savings Booster
6734	Loan	Savings Account	Mid-tier Savings Booster
7265	Credit Card	Savings Account	Mid-tier Savings Booster
1466	Credit Card	Savings Account	Mid-tier Savings Booster
5426	Checking Account	Other	Financial Literacy Program Access
5426	Credit Card	Other	Financial Literacy Program Access
2685	Loan	Savings Account	Premium Investment Services

## Query 8: Currency and FX Exposure Analysis

**Purpose:** Shows the volume of foreign currency transactions by product category.

**Business Use:** Informs FX risk management and helps design tailored products for high-value FX clients (e.g., international credit cards, forex savings).

-- 8. Currency and FX Exposure Analysis

```

SELECT
    Currency,
    ProductCategory,
    COUNT(*) AS TransactionCount,
    SUM(Amount) AS TotalAmount
FROM
    bank_dataset
GROUP BY Currency, ProductCategory
order by ProductCategory;

```

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	Currency	ProductCategory	TransactionCount	TotalAmount			
▶	EUR	Checking Account	3300	12798562.059845643			
	USD	Checking Account	588	2352503.2926533422			
	EUR	Credit Card	3462	17169054.72787779			
	USD	Credit Card	620	3032365.7421194697			
	EUR	Loan	3380	21371390.295252115			
	USD	Loan	618	3880385.2921544863			
	EUR	Mortgage	3402	25794331.187751744			
	USD	Mortgage	588	4507799.517119228			
	EUR	Savings Account	3430	8583706.463934284			
	USD	Savings Account	612	1517003.7710695814			

## Query 9: Product Category Profitability

**Purpose:** Breaks down how much fee revenue each product and subcategory contributes.

**Business Use:** Identifies high-margin products (like premium cards or loans) to guide marketing, cross-selling, and product development.

-- 9. Product Category Profitability

SELECT

ProductCategory,

ProductSubcategory,

SUM(CreditCardFees + InsuranceFees + LatePaymentAmount) AS TotalFees

FROM

bank\_dataset

GROUP BY

ProductCategory, ProductSubcategory

ORDER BY

TotalFees DESC;

Result Grid			
	Filter Rows:	Export:	Wrap Cell Content:
	ProductCategory	ProductSubcategory	TotalFees
▶	Loan	Business	56051.270000000002
	Loan	Gold	52680.030000000006
	Loan	Student	52375.150000000001
	Loan	Standard	51074.150000000002
	Loan	Platinum	50122.619999999999
	Credit Card	Business	36413
	Credit Card	Student	36104
	Credit Card	Gold	35282
	Credit Card	Platinum	34654
	Credit Card	Standard	31253
	Checking Account	Business	16846

## Query 10: Detect Income vs Spending Mismatch (Potential Over-Leverage)

**Purpose:** Detects customers spending more than double their income regularly.

**Business Use:** Useful for credit risk teams to monitor over-leveraged clients and prevent loan defaults.

-- Query 10: Detect Income vs Spending Mismatch (Potential Over-Leverage)

WITH Spending AS (

SELECT

CustomerID,

ROUND(SUM(CASE WHEN TransactionType IN ('Card Payment', 'Withdrawal', 'Transfer', 'Loan Payment') THEN Amount ELSE 0 END), 2) AS TotalSpending,

ROUND(AVG(MonthlyIncome), 2) AS AvgIncome

FROM bank\_dataset

GROUP BY CustomerID

)




SELECT \*,

ROUND(TotalSpending / AvgIncome, 2) AS SpendToIncomeRatio

FROM Spending

WHERE TotalSpending > AvgIncome \* 2

ORDER BY SpendToIncomeRatio DESC;

Result Grid    Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 				
	CustomerID	TotalSpending	AvgIncome	SpendToIncomeRatio
▶	4723	28117.17	2052.2	13.7
	9942	13077.2	1118.54	11.69
	5823	25121.05	2183.07	11.51
	6276	26337.66	2400.48	10.97
	4167	20522.9	1871.29	10.97
	6180	18560.06	1692.96	10.96
	3560	21075.26	1952.55	10.79
	6030	12536.87	1161.98	10.79
	3562	39267.29	3639.98	10.79
	1480	11337.18	1051.49	10.78
	2554	12170.86	1136.58	10.71

Result 39 ×

## Query 11: Geographical Trends in Fee Revenue

**Purpose:** Calculates how much each city contributes to total bank fee revenue.

**Business Use:** Pinpoints locations with high profitability potential and helps prioritize local market strategies.

-- Query 11: Geographical Trends in Fee Revenue

WITH CityFees AS (

SELECT

BranchCity,

ROUND(SUM(CreditCardFees + InsuranceFees + LatePaymentAmount), 2) AS TotalFees

FROM bank\_dataset

GROUP BY BranchCity

),

TotalBankFees AS (

SELECT ROUND(SUM(CreditCardFees + InsuranceFees + LatePaymentAmount), 2) AS BankTotalFees

FROM bank\_dataset

)

SELECT

c.BranchCity,

c.TotalFees,

CONCAT(ROUND(c.TotalFees / t.BankTotalFees \* 100, 2), '%') AS FeeContribution

FROM CityFees c, TotalBankFees t

ORDER BY TotalFees DESC;

Result Grid				Filter Rows:	Export:
	BranchCity	TotalFees	FeeContribution		
▶	Barcelona	85248.43	13.38%		
	Malaga	85090.16	13.35%		
	Murcia	85055.22	13.35%		
	Madrid	80297.15	12.6%		
	Valencia	76144.67	11.95%		
	Bilbao	75778.34	11.89%		
	Zaragoza	75612.81	11.87%		
	Seville	73953.44	11.61%		

## Query 12: Customer Retention Proxy (Months Active)

**Purpose:** Measures the customer's lifespan in the system based on first and last transactions.

**Business Use:** Acts as a proxy for retention and helps identify inactive or dormant accounts needing reactivation campaigns.

-- Query 12: Customer Retention Proxy (Months Active)

WITH CustomerActivity AS (

SELECT

CustomerID,

MIN(DATE(TransactionDate)) AS FirstTxn,

MAX(DATE(TransactionDate)) AS LastTxn

FROM bank\_dataset

GROUP BY CustomerID

)

SELECT

CustomerID,

FirstTxn,

LastTxn,

TIMESTAMPDIFF(MONTH, FirstTxn, LastTxn) AS MonthsActive

FROM CustomerActivity

ORDER BY MonthsActive DESC;



Result Grid					Filter Rows:	Export:	W
	CustomerID	FirstTxn	LastTxn	MonthsActive			
▶	8509	2023-01-12	2025-05-13	28			
	8008	2023-01-04	2025-05-04	28			
	6930	2023-01-11	2025-05-11	28			
	3774	2023-01-11	2025-05-12	28			
	4729	2023-01-02	2025-05-09	28			
	5409	2023-01-07	2025-05-08	28			
	2374	2023-01-09	2025-05-19	28			
	4963	2023-01-04	2025-05-19	28			
	3568	2023-01-01	2025-04-24	27			
	4748	2023-02-11	2025-05-13	27			
	2986	2023-01-17	2025-05-01	27			
Result 41					×		

## Query 14: Product Cross-Sell Score

**Purpose:** Counts how many distinct product categories each customer uses.

**Business Use:** Aids in segmenting multi-product customers (more loyal) and identifying cross-sell opportunities (e.g., upselling insurance to savings users)

-- Query 14: Product Cross-Sell Score

```
WITH ProductCounts AS (
    SELECT CustomerID, COUNT(DISTINCT ProductCategory) AS UniqueProducts
    FROM bank_dataset
    GROUP BY CustomerID
)
SELECT *
FROM ProductCounts
WHERE UniqueProducts >= 3
ORDER BY UniqueProducts DESC;
```

Result Grid			Filter Rows:
	CustomerID	UniqueProducts	
▶	1191	5	
	1201	5	
	1615	5	
	1815	5	
	1845	5	
	1869	5	
	1975	5	
	2316	5	
	2359	5	
	2620	5	
	3122	5	

Result 42 ×

## Query 15: First vs Last Channel Usage (Shift to Digital?)

**Purpose:** Tracks if a customer has shifted from physical to digital channels over time.

**Business Use:** Measures digital transformation progress and helps evaluate the success of mobile/online banking adoption initiatives

-- Query 15: First vs Last Channel Usage (Shift to Digital?)

WITH RankedTxn AS (

SELECT

CustomerID,

Channel,

TransactionDate,

ROW\_NUMBER() OVER (PARTITION BY CustomerID ORDER BY TransactionDate ASC) AS FirstUse,

ROW\_NUMBER() OVER (PARTITION BY CustomerID ORDER BY TransactionDate DESC) AS LastUse

FROM bank\_dataset

),

FirstLast AS (

SELECT

CustomerID,

MAX(CASE WHEN FirstUse = 1 THEN Channel END) AS FirstChannel,

MAX(CASE WHEN LastUse = 1 THEN Channel END) AS LastChannel

FROM RankedTxn

GROUP BY CustomerID

)

SELECT \*

FROM FirstLast

WHERE FirstChannel != LastChannel;

**Result Grid**

Filter Rows:

Export:



	CustomerID	FirstChannel	LastChannel
▶	1001	ATM	Online
	1002	Branch	Online
	1004	ATM	Branch
	1007	Branch	ATM
	1009	Branch	ATM
	1011	Mobile	Online
	1012	Online	Mobile
	1015	Mobile	Branch
	1016	Mobile	Branch
	1018	Mobile	Branch
	1020	Mobile	ATM

Result 43 ×