

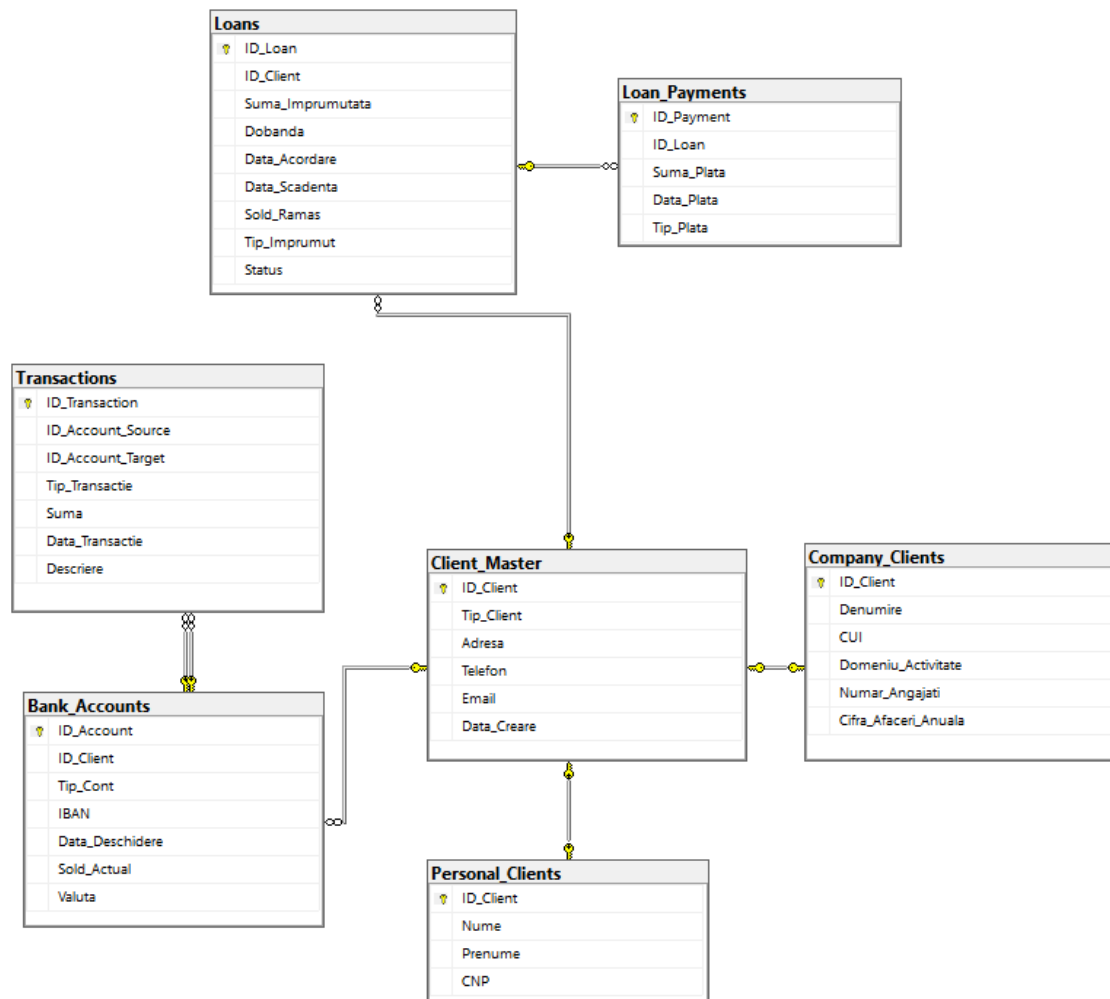
Management Bancar

Descriere

Proiectul permite gestionarea conturilor bancare, efectuarea tranzacțiilor și administrarea împrumuturilor pentru clienții unei bănci. Funcționalitățile includ: crearea și gestionarea conturilor de economii, curente și depozite, efectuarea de tranzacții (depuneri, retrageri, transferuri), acordarea și urmărirea împrumuturilor.

Descrierea bazei de date

- Diagrama



- Structura tabelor

Client_Master

Stochează informații generale despre clienți (persoane fizice și juridice), cum ar fi tipul clientului, adresa, telefonul, emailul și data creării.

Personal_Clients

Stochează informații suplimentare pentru clienții persoane fizice, incluzând numele, prenumele și codul numeric personal (CNP).

Company_Clients

Stochează informații suplimentare pentru clienții persoane juridice, incluzând denumirea companiei, codul unic de identificare (CUI), domeniul de activitate, numărul de angajați și cifra de afaceri anuală.

Bank_Accounts

Conține detalii despre conturile bancare ale clienților, cum ar fi tipul contului (curent, economii, depozit), IBAN-ul, data deschiderii, soldul actual și valuta contului.

Transactions

Păstrează un istoric al tranzacțiilor efectuate între conturi, inclusiv tipul tranzacției (depunere, retragere, transfer), suma implicată, data tranzacției și o descriere opțională.

Loans

Stochează detalii despre împrumuturile acordate clienților, incluzând suma împrumutată, dobânda, datele de acordare și scadență, soldul rămas, tipul împrumutului (persoană fizică sau juridică) și starea acestuia (activ, finalizat, restant).

Loan_Payments

Conține informații despre plățile efectuate pentru împrumuturi, incluzând suma plătită, data plății și tipul plății (normală sau penalizare).

- Descrierea constrângerilor de integritate

1. Chei primare (Primary Keys)

Fiecare tabel are o cheie primară care garantează unicitatea fiecărui rând din tabel:

- Client_Master(ID_Client)
- Personal_Clients(ID_Client)
- Company_Clients(ID_Client)
- Bank_Accounts(ID_Account)
- Transactions(ID_Transaction)
- Loans(ID_Loan)
- Loan_Payments(ID_Payment)

2. Chei externe (Foreign Keys)

Cheile externe definesc relațiile între tabele, asigurând referința corectă între entități:

- Personal_Clients(ID_Client) și Company_Clients(ID_Client) ↔ Client_Master(ID_Client).
- Bank_Accounts(ID_Client) ↔ Client_Master(ID_Client).
- Transactions(ID_Account_Source), Transactions(ID_Account_Target) ↔ Bank_Accounts(ID_Account).
- Loans(ID_Client) ↔ Client_Master(ID_Client).
- Loan_Payments(ID_Loan) ↔ Loans(ID_Loan).

3. Constrângeri de unicitate (Unique Constraints)

Aceste constrângeri asigură faptul că anumite valori sunt unice în baza de date:

- Client_Master(Email) asigură unicitatea adresei de email.
- Personal_Clients(CNP) asigură unicitatea codului numeric personal.
- Company_Clients(CUI) asigură unicitatea codului unic de identificare.
- Bank_Accounts(IBAN) garantează unicitatea codului IBAN.

4. Valori implicite și constrângeri CHECK

- **Tipuri predefinite de date:**
 - Client_Master(Tip_Client) permite doar valorile: 'persoana fizica', 'persoana juridica'.
 - Bank_Accounts(Tip_Cont) permite doar valorile: 'curent', 'economii', 'depozit'.
 - Bank_Accounts(Valuta) permite doar valorile: 'RON', 'EUR', 'USD'.
 - Transactions(Tip_Transactie) permite doar valorile: 'depunere', 'retragere', 'transfer'.
 - Loans(Tip_Imprumut) permite doar valorile: 'persoana fizica', 'persoana juridica'.
 - Loans(Status) permite doar valorile: 'activ', 'finalizat', 'restant'.
 - Loan_Payments(Tip_Plata) permite doar valorile: 'normala', 'penalizare'.
- **Domenii de valori numerice:**
 - Bank_Accounts(Sold_Actual) impune ca valoarea să fie pozitivă sau egală cu zero.
 - Loans(Suma_Imprumutata) și Loans(Sold_Ramas) trebuie să fie valori pozitive sau egale cu zero.
 - Loans(Dobanda) trebuie să fie pozitivă.
 - Transactions(Suma) impune o valoare pozitivă.
- **Condiții suplimentare:**
 - Company_Clients(Numar_Angajati) trebuie să fie mai mare decât 0.

5. Integritatea referențială și acțiuni pe ștergere/actualizare

Pentru cheile externe, se definesc acțiuni ce păstrează integritatea referențială:

- Ștergerea unui client din Client_Master șterge automat toate înregistrările din Personal_Clients, Company_Clients, Bank_Accounts și Loans asociate (ON DELETE CASCADE).
- Actualizarea unui ID_Client sau ID_Loan propagă modificarea în tabelele dependente (ON UPDATE CASCADE).

Descrierea procedurilor și funcțiilor

Am implementat 3 view-uri pentru a facilita analiza datelor și generarea rapoartelor în sistemul bancar. Acestea sunt utilizate pentru a structura și agrega informațiile din baza de date într-un mod util pentru procesul decizional. Fiecare view răspunde unor întrebări specifice, adresând nevoile operaționale și de analiză ale băncii.

Complexitate 7

```
CREATE OR ALTER VIEW View_Analiza_Performanta_Conturi_Tranzactii AS
SELECT
    BA.ID_Client,
    CASE
        WHEN CM.Tip_Client = 'persoana fizica' THEN CONCAT(PC.Prenume, ' ', PC.Nume)
        ELSE CC.Denumire
    END AS Nume_Or_Denumire,
    BA.Tip_Cont,
    COUNT(DISTINCT T.ID_Transaction) AS Numar_Tranzactii,
    SUM(CASE WHEN T.Tip_Transactie = 'depunere' THEN T.Suma ELSE 0 END) AS Total_Depuneri,
    SUM(CASE WHEN T.Tip_Transactie = 'retragere' THEN T.Suma ELSE 0 END) AS Total_Retrageri,
    SUM(CASE WHEN T.Tip_Transactie = 'transfer' THEN T.Suma ELSE 0 END) AS Total_Transferuri,
    SUM(BA.Sold_Actual) AS Sold_Actual,
    COUNT(DISTINCT BA.ID_Account) AS Numar_Conturi
FROM Bank_Accounts BA
JOIN Client_Master CM ON BA.ID_Client = CM.ID_Client
LEFT JOIN Personal_Clients PC ON CM.ID_Client = PC.ID_Client
LEFT JOIN Company_Clients CC ON CM.ID_Client = CC.ID_Client
LEFT JOIN Transactions T ON BA.ID_Account IN (T.ID_Account_Source, T.ID_Account_Target)
WHERE BA.Sold_Actual > 0
GROUP BY
    BA.ID_Client,
    CM.Tip_Client,
    BA.Tip_Cont,
    PC.Nume,
    PC.Prenume,
    CC.Denumire
HAVING
    COUNT(DISTINCT T.ID_Transaction) > 1
    AND SUM(BA.Sold_Actual) > 20000;

SELECT * FROM View_Analiza_Performanta_Conturi_Tranzactii;
```

Acest view oferă o imagine detaliată a activității clienților, permițând băncii să identifice clienții valoroși care au tranzacții frecvente și solduri mari, să analizeze tiparele de utilizare a conturilor (curente, de economii, depozite) și să monitorizeze tipurile de tranzacții efectuate (depuneri, retrageri, transferuri).

Complexitate 6

```
CREATE OR ALTER VIEW View_Bilant_Anual_Complex AS
SELECT
    YEAR(T.Data_Transactie) AS An,
    BA.Tip_Cont,
    COUNT(DISTINCT T.ID_Transaction) AS Numar_Tranzactii,
    SUM(CASE WHEN T.Tip_Transactie = 'depunere' THEN T.Suma ELSE 0 END) AS Total_Depuneri,
    SUM(CASE WHEN T.Tip_Transactie = 'retragere' THEN T.Suma ELSE 0 END) AS Total_Retrageri,
    SUM(LP.Suma_Plata) AS Total_Penalizari_Collectate,
    AVG(LP.Suma_Plata) AS Penalizare_Medie,
    SUM(L.Dobanda / 100 * L.Suma_Imprumutata) AS Venituri_Din_Dobanzi,
    AVG(L.Dobanda) AS Dobanda_Medie,
    SUM(BA.Sold_Actual) AS Sold_Total_Conturi,
    SUM(L.Suma_Imprumutata) AS Total_Imprumuturi_Acordate,
    SUM(BA.Sold_Actual) - SUM(L.Suma_Imprumutata) AS Diferenta_Active_Pasive
FROM Bank_Accounts BA
LEFT JOIN Transactions T ON BA.ID_Account = T.ID_Account_Source OR BA.ID_Account = T.ID_Account_Target
LEFT JOIN Loans L ON BA.ID_Client = L.ID_Client
LEFT JOIN Loan_Payments LP ON L.ID_Loan = LP.ID_Loan
WHERE YEAR(T.Data_Transactie) = 2023
    OR YEAR(LP.Data_Plata) = 2023
GROUP BY YEAR(T.Data_Transactie), BA.Tip_Cont
HAVING SUM(LP.Suma_Plata) > 1000;

SELECT * FROM View_Bilant_Anual_Complex;
```

Acest view oferă o analiză detaliată a bilanțului anual pentru conturile băncii, incluzând numărul de tranzacții, sumele totale pentru depuneri, retrageri și penalizări colectate, precum și informații despre dobânzi și active versus pasive. Se calculează media penalizărilor și dobânzilor, totalul sumelor acordate sub formă de împrumuturi și diferența între activele și pasivele băncii.

Complexitate 6

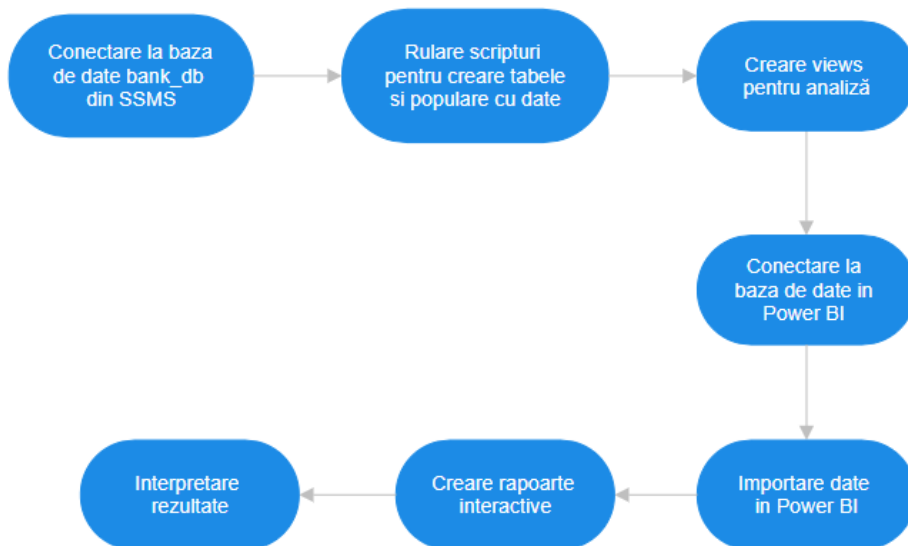
```
CREATE OR ALTER VIEW View_Informatii_Clienti AS
SELECT
    CM.ID_Client,
    CM.Tip_Client,
    CASE
        WHEN CM.Tip_Client = 'persoana fizica' THEN CONCAT(PC.Prenume, ' ', PC.Nume)
        ELSE CC.Denumire
    END AS Nume_Or_Denumire,
    CM.Adresa,
    CM.Telefon,
    CM.Email,
    COUNT(DISTINCT BA.ID_Account) AS Numar_Conturi_Bancare,
    SUM(BA.Sold_Actual) AS Sold_Total_Conturi,
    COUNT(DISTINCT L.ID_Loan) AS Numar Imprumuturi,
    SUM(L.Sold_Ramas) AS Sold_Total Imprumuturi,
    SUM(CASE WHEN LP.Tip_Plata = 'penalizare' THEN LP.Suma_Plata ELSE 0 END) AS Total_Penalizari
FROM Client_Master CM
LEFT JOIN Personal_Clients PC ON CM.ID_Client = PC.ID_Client
LEFT JOIN Company_Clients CC ON CM.ID_Client = CC.ID_Client
LEFT JOIN Bank_Accounts BA ON CM.ID_Client = BA.ID_Client
LEFT JOIN Loans L ON CM.ID_Client = L.ID_Client
LEFT JOIN Loan_Payments LP ON L.ID_Loan = LP.ID_Loan
GROUP BY
    CM.ID_Client, CM.Tip_Client, CM.Adresa, CM.Telefon, CM.Email, PC.Nume, PC.Prenume, CC.Denumire;

SELECT * FROM View_Informatii_Clienti;
```

Acest view oferă o perspectivă completă asupra clienților băncii, incluzând informații de identificare (nume, adresă, telefon, email), activitatea financiară (numărul de conturi bancare, soldul total al conturilor), precum și obligațiile lor față de bancă (numărul de împrumuturi, soldul total rămas și penalizările acumulate).

Descrierea aplicației

Fluxul de lucru



Fluxul aplicației pentru gestionarea datelor băncii începe cu conectarea la baza de date bank_db utilizând SQL Server Management Studio (SSMS). Ulterior, sunt rulate scripturile necesare pentru crearea tabelor și popularea acestora cu date. Se creează apoi views pentru analiza datelor, care oferă o structură predefinită pentru rapoarte. După această etapă, datele sunt conectate și importate în Power BI pentru procesarea și vizualizarea interactivă. În Power BI, sunt generate rapoarte dinamice care

permit interpretarea detaliată a rezultatelor, sprijinind luarea deciziilor informate.

Conexiunea la baza de date

Login Connection Properties Always Encrypted Additional Connection Pa

Server

Server type: Database Engine

Server name: localhost:1433

Authentication: SQL Server Authentication

Login: SA

Password: *****

☒ Remember password

Connection Security

Encryption: Mandatory

☒ Trust server certificate

Host name in certificate:

Connect Cancel Help

Bază de date SQL Server

Server: localhost:1433

Bază de date (opțional): bank_db

Modul Conectivitate de date ⓘ

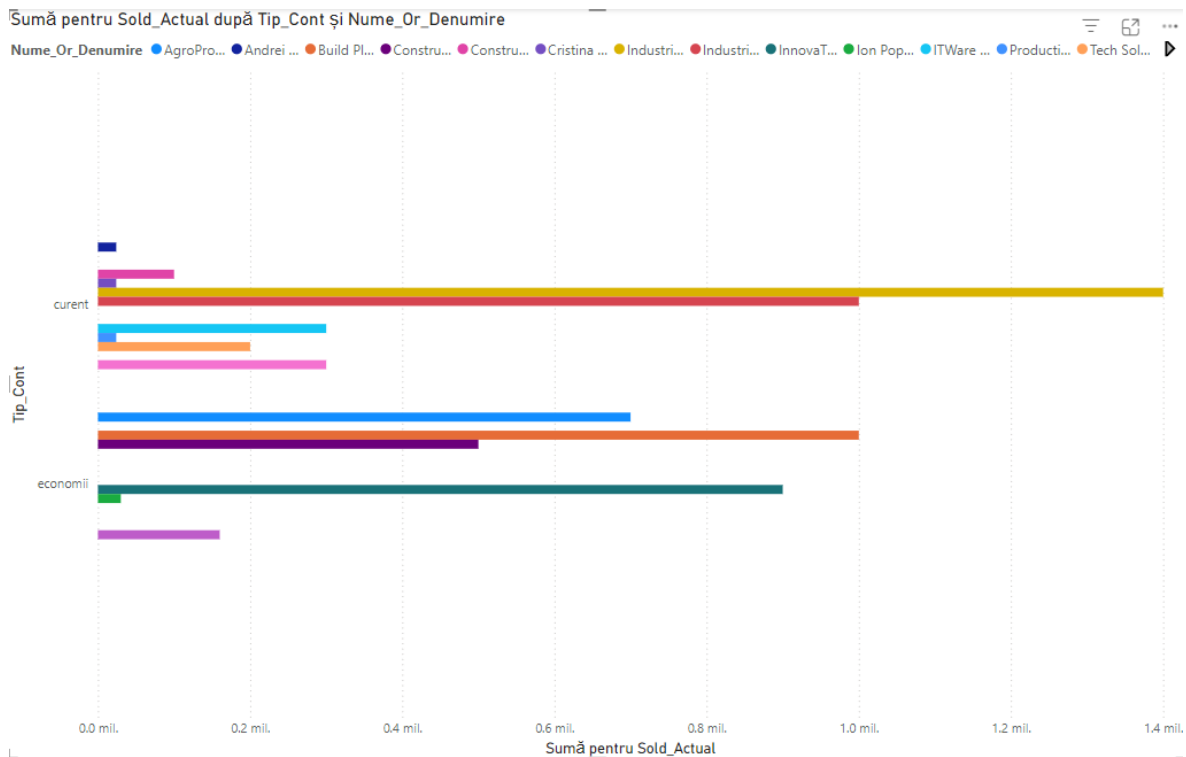
☒ Import

☐ DirectQuery

Opțiuni complexe

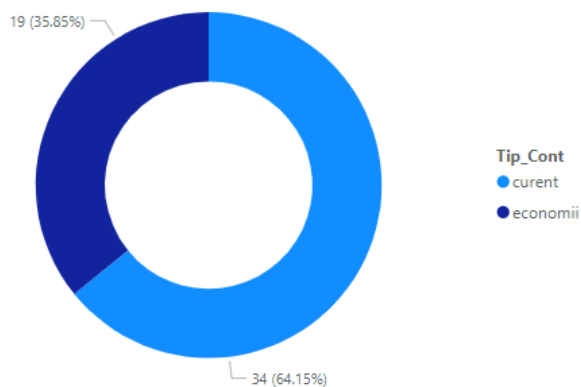
OK

Rapoarte interactive



Scopul principal al acestui raport este să ofere o imagine clară asupra modului în care clienții utilizează diferitele tipuri de conturi și care sunt cei mai valoroși clienți din punct de vedere al soldurilor.

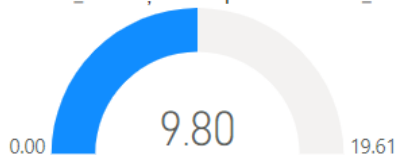
Sumă pentru Numar_Transactii după Tip_Cont



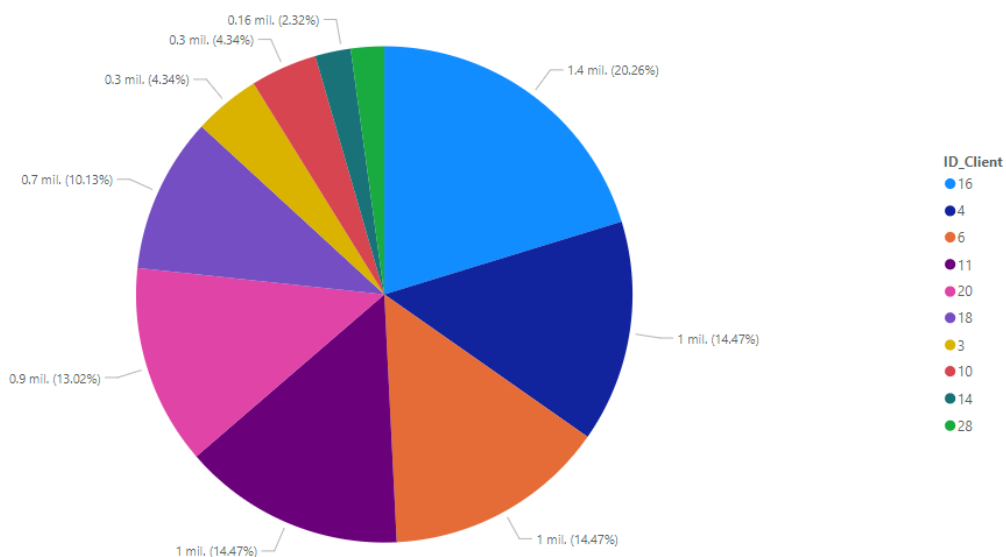
Primul grafic reprezintă distribuția numărului de tranzacții pe tipuri de conturi (curent și economii). Este util pentru a înțelege care tip de cont generează cele mai multe tranzacții.

Al doilea grafic oferă o vedere asupra dobânzii medii aplicate împrumuturilor și veniturilor generate din dobânzi. Ajută banca să monitorizeze eficiența politicilor de creditare și să optimizeze strategia de acordare a împrumuturilor.

Sumă pentru Dobanda_Medie și Sumă pentru Venituri_Din_Dobanzi



Sumă pentru Sold_Total_Conturi după ID_Client



Arată care clienți dețin cele mai mari solduri în conturile lor, evidențiind astfel clienții cu valoare ridicată pentru bancă. Acest raport este util pentru departamentele financiare și de marketing ale băncii, permițându-le să aloce resurse mai eficient și să îmbunătățească relațiile cu clienții cheie.

Concluzie

Acest proiect a fost o soluție practică și bine integrată pentru gestionarea activităților unei bănci. Am construit o bază de date care permite administrarea conturilor bancare, monitorizarea tranzacțiilor și urmărirea împrumuturilor într-un mod organizat și eficient. Pe lângă aceasta, am creat rapoarte complexe care oferă o imagine clară asupra activităților clienților și a performanței băncii.

Prin vizualizările și analizele generate, banca poate identifica clienți valoroși, poate înțelege mai bine tiparele financiare și poate lua decizii informate, bazate pe date reale. Integrarea cu Power BI a adăugat un plus de interactivitate, făcând informațiile mai ușor de interpretat și mai accesibile pentru utilizatori.

În final, proiectul a demonstrat cum o structură bine gândită a bazei de date, combinată cu instrumente de vizualizare moderne, poate sprijini luarea deciziilor și îmbunătăți modul în care banca își desfășoară activitatea. Este o soluție care ajută nu doar la organizare, ci și la dezvoltarea unor strategii mai eficiente pentru creșterea satisfacției clienților și optimizarea operațiunilor.

Bibliografie

<https://ocw.cs.pub.ro/courses/bdd/laboratoare/09>

<https://learn.microsoft.com/en-us/power-bi/>

<https://mockaroo.com/> (pentru generare de date)