

**Here’s a comprehensive prompt you can use to generate a synthetic banking dataset in SQL Server, tailored for Power BI dashboarding and data cleaning training. This prompt includes every requirement with example specifications for each, recommendations for KPIs, visuals, and DAX measures.**

Dataset Generation Prompt  
I am building a banking dataset in SQL Server for Power BI reporting and data cleaning practice. Please generate and insert approximately 10,000 records across the following tables and columns:  
Table Structure Example  
Accounts: AccountID, CustomerID, AccountType, OpenDate, BranchID, Balance  
Transactions: TransactionID, AccountID, TransactionDate, Amount, TransactionType, Description  
Customers: CustomerID, Name, Gender, DateOfBirth, Address, City, State, Zip, Email, Phone, Occupation, CreatedDate, Status  
Insert Data Generation Guidelines  
Use realistic bank-y data ranges. For transactions, include deposits, withdrawals, charges/fees, and transfers.  
Include deliberate data quality issues, e.g.:  
Duplicate records in Customers and/or Transactions  
Inconsistent formatting for text fields (e.g., emails like "[JSmith@Email.com](mailto:JSmith@Email.com)", "[jsmith@email.com](mailto:jsmith@email.com)", phone numbers as "(555)123-4567", "555 123 4567", "5551234567", etc.)  
Null or blank values in non-critical columns (e.g., missing emails, addresses)  
Random mislabels: e.g., AccountType as 'Current', 'Saving', 'Savings Account', etc., or gender as 'M', 'Male', 'F', 'Female', 'FEMALE'  
Out-of-range or negative values in transaction Amount or Account Balance, to teach error discovery  
Typos in City or Occupation (e.g., "bankar", "banker", "managar", "manager")  
Incorrect dates (future-dated transactions, duplicate TransactionID, mismatched AccountID)  
Example for Data Cleaning Issues  
text

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CustomerID | Name | Email | Gender | Occupation |
| 1001 | Jane Doe | [jane@x.com](mailto:jane@x.com) | F | Banker |
| 1002 | John Doe | [JDOE@X.COM](mailto:JDOE@X.COM) | MALE | Banakr |
| 1003 | Jane Doe | [jane@x.com](mailto:jane@x.com) | Female | Banker | <-- duplicate & inconsistent |
| 1004 |  | n/a |  | Manager | <-- missing values |

KPIs and Power BI Report Requirements  
Your dataset should support at least 10 KPIs, enabling two Power BI report pages (5 KPIs per page). Also, provide recommended visuals and DAX measure examples.  
10 Suggested Banking KPIs  
KPI NameBrief DescriptionVisual RecommendationDAX Measure Example

1. Total Deposits  
   Sum of deposit transactions  
   Card + Trend Line  
   Total Deposits = CALCULATE(SUM(Transactions[Amount]), Transactions[TransactionType]="Deposit")
2. Total Withdrawals  
   Sum of withdrawal transactions  
   Card + Trend Line  
   Total Withdrawals = CALCULATE(SUM(Transactions[Amount]), Transactions[TransactionType]="Withdrawal")
3. Net Interest Earned  
   Interest income minus expense  
   Card + Stacked Column  
   Net Interest = SUM(Transactions[Amount]) - SUM(Transactions[InterestExpense]) (if fields available)
4. Average Account Balance  
   Mean of current balances  
   Gauge / Card  
   Avg Balance = AVERAGE(Accounts[Balance])
5. Customer Churn Rate  
   % customers with closed accounts  
   Donut/Pie  
   Churn Rate = DIVIDE(CALCULATE(COUNTROWS(Customers), Customers[Status]="Closed"), COUNTROWS(Customers))
6. Loan-to-Deposit Ratio  
   Total loans divided by total deposits  
   KPI Visual (Indicator)  
   LDR = DIVIDE(SUM(Accounts[LoanBalance]), [Total Deposits])
7. Cost-to-Income Ratio  
   Expenses divided by revenue  
   Gauge/Stacked Column  
   CTI = DIVIDE(SUM(Expenses[Amount]), SUM(Income[Amount]))
8. Return on Equity (ROE)  
   Profit as % of equity  
   Card + Clustered Column  
   ROE = DIVIDE([Net Profit], SUM(Accounts[Equity]))
9. Active Customers Count  
   Number of active customer accounts  
   Card + Bar  
   Active Customers = CALCULATE(COUNTROWS(Customers), Customers[Status]="Active")
10. NPA Ratio  
    Non-performing assets / total assets  
    Clustered Column/Bar  
    NPA Ratio = DIVIDE(SUM(Accounts[NPA]), SUM(Accounts[Balance]))  
    Adjust measures as per final table schema. Add ‘Amount’ fields as relevant.  
    Example DAX Syntax for Power BI (with explanations)  
    text  
    -- Total Deposits  
    Total Deposits =  
    CALCULATE(  
    SUM(Transactions[Amount]),  
    Transactions[TransactionType] = "Deposit"  
    )

-- Customer Churn Rate  
Churn Rate =  
DIVIDE(  
CALCULATE(COUNTROWS(Customers), Customers[Status] = "Closed"),  
COUNTROWS(Customers)  
)

Adjust to suit your actual column names and logic.  
Power BI Visuals Recommendations  
Cards: For headline numbers (e.g., deposits, withdrawals, balances)  
KPI Visuals: To show performance vs. targets/trends  
Donut/Pie: Customer churn, product adoption  
Bar/Column charts: Account types, transactions over time, customer segments  
Gauge: Ratios (loan-to-deposit, cost-to-income)  
Example Prompt Ready for Use  
Generate T-SQL insert scripts for a banking dataset with ~10,000 rows spanning Customers, Accounts, Transactions. Include realistic banking fields (see above), and enforce data issues such as duplicates, inconsistent formats, typos, nulls, and outliers (see examples provided). Ensure the dataset is suitable for Power BI KPI dashboarding using SQL Server as source. Also provide a summary table of 10 possible KPIs (with chart/visual and DAX suggestions—see table above) that can be split evenly across two Power BI report pages.  
This prompt and example structure will let you create a feature-rich, purposely “dirty” banking database for Power BI learning, complete with practical KPIs and reporting guidance

Here’s a complete T-SQL generation prompt, plus ready-to-use guidance for creating a synthetic, “dirty” banking dataset and building Power BI dashboards with meaningful KPIs.

**Banking Dataset Generation Prompt (for SQL Server)**

**Objective:**  
Create a realistic banking dataset (~10,000 rows) for Power BI dashboarding and data cleaning practice. Embed data quality issues to allow for training in error detection and cleaning.

**1. Table Structures and Example Columns**

-- Customers Table  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 Name NVARCHAR(100),  
 Gender NVARCHAR(20),  
 DateOfBirth DATE,  
 Address NVARCHAR(200),  
 City NVARCHAR(100),  
 State NVARCHAR(50),  
 Zip NVARCHAR(20),  
 Email NVARCHAR(100),  
 Phone NVARCHAR(30),  
 Occupation NVARCHAR(100),  
 CreatedDate DATE,  
 Status NVARCHAR(20)  
);  
  
-- Accounts Table  
CREATE TABLE Accounts (  
 AccountID INT PRIMARY KEY,  
 CustomerID INT,  
 AccountType NVARCHAR(50),  
 OpenDate DATE,  
 BranchID INT,  
 Balance DECIMAL(18,2)  
);  
  
-- Transactions Table  
CREATE TABLE Transactions (  
 TransactionID INT PRIMARY KEY,  
 AccountID INT,  
 TransactionDate DATE,  
 Amount DECIMAL(18,2),  
 TransactionType NVARCHAR(30),  
 Description NVARCHAR(500)  
);

**2. Data Generation and Data Issues Guidance**

* Generate ~2,000 unique customers, ~3,000 accounts, and 10,000+ transactions.
* Insert common data issues such as:
  + **Duplicate records** in Customers and Transactions.
  + **Inconsistent formatting**: e.g., [JSmith@Email.com], [jsmith@email.com], phone as "(555)123-4567", "5551234567".
  + **Null or blank** fields (missing emails/phones/addresses/genders).
  + **Typos/mislabels**: e.g., "Bankar", "managar", "Current", "Savings Account".
  + **Out-of-range/negative values** for Amount, Balance.
  + **Gender/AccountType labeling inconsistencies**: "M", "Male", "male", "FEMALE", "f".
  + **Incorrect dates**: future-dated transactions, duplicate TransactionID, mismatched AccountID.
  + Additional plausible errors for data cleaning.

**Example Insert with Deliberate Errors:**

INSERT INTO Customers VALUES (1001, 'Jane Doe', 'F', '1985-03-20', '123 Main St', 'New York', 'NY', '10001', 'jane@x.com', '(555)123-4567', 'Banker', '2010-02-12', 'Active');  
INSERT INTO Customers VALUES (1002, 'John Doe', 'male', '1979-11-15', '124 Main St', 'Newark', 'NJ', '07102', 'JDOE@X.COM', '555 321 9876', 'Banakr', '2009-04-10', 'active');  
INSERT INTO Customers VALUES (1003, 'Jane Doe', 'Female', '1985-03-20', '123 Main St', 'New York', 'NY', '10001', 'jane@x.com', '5551234567', 'Banker', '2010-02-12', 'Active'); -- Duplicate  
INSERT INTO Customers VALUES (1004, '', '', NULL, NULL, 'Los Angles', 'CA', '', 'n/a', NULL, 'Manager', NULL, 'Closed'); -- Missing/typo fields

Continue similar inserts and script generation using CTEs, random sampling, or tools like SQL Data Generator.

**3. KPIs, Power BI Visuals, DAX Measures**

|  |  |  |  |
| --- | --- | --- | --- |
| KPI Name | Brief Description | Example Visual | Example DAX Measure |
| Total Deposits | Sum of deposit transactions | Card + Trend Line | Total Deposits = CALCULATE(SUM(Transactions[Amount]), Transactions[TransactionType]="Deposit") |
| Total Withdrawals | Sum of withdrawal transactions | Card + Trend Line | Total Withdrawals = CALCULATE(SUM(Transactions[Amount]), Transactions[TransactionType]="Withdrawal") |
| Net Interest Earned | Interest income minus expense | Card + Stacked Column | Net Interest = SUM(Transactions[Amount]) - SUM(Transactions[InterestExpense]) |
| Average Account Balance | Mean of balances | Gauge / Card | Avg Balance = AVERAGE(Accounts[Balance]) |
| Customer Churn Rate | % customers with closed accounts | Donut/Pie | Churn Rate = DIVIDE(CALCULATE(COUNTROWS(Customers), Customers[Status]="Closed"), COUNTROWS(Customers)) |
| Loan-to-Deposit Ratio | Loans divided by deposits | KPI Visual | LDR = DIVIDE(SUM(Accounts[LoanBalance]), [Total Deposits]) |
| Cost-to-Income Ratio | Expenses divided by revenue | Gauge/Bar | CTI = DIVIDE(SUM(Expenses[Amount]), SUM(Income[Amount])) |
| Return on Equity (ROE) | Profit as % of equity | Card + Column | ROE = DIVIDE([Net Profit], SUM(Accounts[Equity])) |
| Active Customers Count | Number of active customers | Card + Bar | Active Customers = CALCULATE(COUNTROWS(Customers), Customers[Status]="Active") |
| NPA Ratio | Non-performing assets/total assets | Clustered Bar | NPA Ratio = DIVIDE(SUM(Accounts[NPA]), SUM(Accounts[Balance])) |

Split these evenly (5 per page) across two Power BI report pages.

**4. Power BI Visual Recommendations**

* **Cards:** For headline KPIs (deposits, withdrawals, balances).
* **KPI Visuals:** Show progress/trends vs. targets.
* **Donut/Pie:** Display churn, account types, or status.
* **Bar/Column:** For segments (branch, city, occupation).
* **Gauge:** Ratios (loan-to-deposit, cost-to-income).

**5. Example DAX Measure Syntax**

-- Total Deposits  
Total Deposits =   
CALCULATE(  
 SUM(Transactions[Amount]),  
 Transactions[TransactionType] = "Deposit"  
)  
  
-- Customer Churn Rate  
Churn Rate =   
DIVIDE(  
 CALCULATE(COUNTROWS(Customers), Customers[Status] = "Closed"),  
 COUNTROWS(Customers)  
)

Adjust logic as needed for your model and dataset.

This prompt and structure ensure your SQL Server database will produce a realistic, training-ready banking dataset—perfect for both Power BI data modeling and data quality learning.

**write a sql query that will help create requireed tables and then write sql queries that will insert data into that table**

Here are the SQL queries to both **create the required tables** for your banking dataset and **insert sample rows** (including deliberate data issues for data cleaning training):

**1. Create Required Tables**

-- Customers Table  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 Name NVARCHAR(100),  
 Gender NVARCHAR(20),  
 DateOfBirth DATE,  
 Address NVARCHAR(200),  
 City NVARCHAR(100),  
 State NVARCHAR(50),  
 Zip NVARCHAR(20),  
 Email NVARCHAR(100),  
 Phone NVARCHAR(30),  
 Occupation NVARCHAR(100),  
 CreatedDate DATE,  
 Status NVARCHAR(20)  
);  
  
-- Accounts Table  
CREATE TABLE Accounts (  
 AccountID INT PRIMARY KEY,  
 CustomerID INT,  
 AccountType NVARCHAR(50),  
 OpenDate DATE,  
 BranchID INT,  
 Balance DECIMAL(18,2)  
);  
  
-- Transactions Table  
CREATE TABLE Transactions (  
 TransactionID INT PRIMARY KEY,  
 AccountID INT,  
 TransactionDate DATE,  
 Amount DECIMAL(18,2),  
 TransactionType NVARCHAR(30),  
 Description NVARCHAR(500)  
);

**2. Example SQL Insert Queries (Including Data Issues)**

Below are insertion examples for each table, embedding typical data issues that are useful for data cleaning practice in Power BI:

-- Customers with duplicates, inconsistent gender, formatting, and missing data  
INSERT INTO Customers VALUES (1001, 'Jane Doe', 'F', '1985-03-20', '123 Main St', 'New York', 'NY', '10001', 'jane@x.com', '(555)123-4567', 'Banker', '2010-02-12', 'Active');  
INSERT INTO Customers VALUES (1002, 'John Doe', 'MALE', '1979-11-15', '124 Main St', 'Newark', 'NJ', '07102', 'JDOE@X.COM', '555 321 9876', 'Banakr', '2009-04-10', 'active');  
INSERT INTO Customers VALUES (1003, 'Jane Doe', 'Female', '1985-03-20', '123 Main St', 'New York', 'NY', '10001', 'jane@x.com', '5551234567', 'Banker', '2010-02-12', 'Active'); -- Duplicate  
INSERT INTO Customers VALUES (1004, '', '', NULL, NULL, 'Los Angles', 'CA', '', 'n/a', NULL, 'Manager', NULL, 'Closed'); -- Missing/typo fields  
  
-- Accounts with inconsistent AccountType and out-of-range balances  
INSERT INTO Accounts VALUES (2001, 1001, 'Savings Account', '2015-03-01', 301, 5342.55);  
INSERT INTO Accounts VALUES (2002, 1002, 'Current', '2018-07-15', 302, 12500.00);  
INSERT INTO Accounts VALUES (2003, 1003, 'Saving', '2019-11-23', 301, -155.25); -- Negative balance  
INSERT INTO Accounts VALUES (2004, 1004, 'Current', '2026-01-01', 303, 400.00); -- Future OpenDate  
  
-- Transactions with formatting, outliers, duplication, missing fields  
INSERT INTO Transactions VALUES (3001, 2001, '2024-01-10', 200.00, 'Deposit', 'Monthly Salary');  
INSERT INTO Transactions VALUES (3002, 2002, '2024-02-11', -100.00, 'Withdrawal', 'ATM Cash');  
INSERT INTO Transactions VALUES (3003, 2002, '2023-11-20', 0.00, 'Deposit', '---'); -- Zero amount, meaningless desc  
INSERT INTO Transactions VALUES (3004, 2003, '2028-05-15', 250.00, 'deposit', 'Interest Credit'); -- Future date, inconsistent TransactionType  
INSERT INTO Transactions VALUES (3005, 2004, '2024-04-01', 150.00, 'Fee', NULL); -- Null Description  
INSERT INTO Transactions VALUES (3002, 2002, '2024-02-11', -100.00, 'Withdrawal', 'ATM Cash'); -- Duplicate TransactionID

**Notes:**

* To create a much larger dataset, you can automate the data generation using a script or a data generation tool, but the above structure (with deliberate formatting and quality errors) is what you should use as a template.
* Make sure to **adjust the range and variability** when scaling to thousands of rows, including randomization, duplication, typos, inconsistent capitalizations, and missing fields as needed for training.
* Use these inserts and structures as a starting sample for data model building and cleaning in Power BI.

If you need an automated script for bulk data generation, let me know!

**i dont think it will create dataset with 10K records**

You are correct—the sample SQL insert statements previously provided were intended as examples to illustrate table design and what “dirty” or inconsistent data would look like, not to generate the full 10,000-record dataset. Manually writing 10,000 individual INSERT statements is not practical.

**How to Generate a Realistic 10,000-Record Dataset**

**1. Use T-SQL Loops and Functions**

You can use T-SQL scripts with loops (e.g., via a WHILE loop) and random data generators (like NEWID(), RAND(), DATEADD(), etc.) to bulk-insert thousands of rows with varied data and intentional data issues. Below is a simplified example for the Customers table:

-- Example: Generate 10,000 Customers (simplified; for robust variety, expand logic)  
SET NOCOUNT ON;  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Customers (  
 CustomerID, Name, Gender, DateOfBirth, Address, City, State, Zip, Email, Phone, Occupation, CreatedDate, Status  
 )  
 VALUES (  
 @i,  
 CONCAT('Name', @i),  
 CASE WHEN @i % 5 = 0 THEN 'Male' WHEN @i % 5 = 1 THEN 'F' ELSE 'Female' END, -- Gender variety  
 DATEADD(DAY, - (20 + (@i % 40)) \* 365, GETDATE()),  
 CONCAT(@i, ' Main St'),  
 CASE WHEN @i % 20 = 0 THEN 'bankar' ELSE 'New York' END, -- Typos  
 'State',  
 RIGHT('0000' + CAST(@i AS VARCHAR(4)), 5),  
 CONCAT('user', @i, CASE WHEN @i % 10 = 0 THEN '@EMAIL.com' ELSE '@email.com' END), -- Inconsistent email  
 CASE WHEN @i % 7 = 0 THEN '' ELSE CONCAT('555', @i) END, -- Missing phone  
 CASE WHEN @i % 50 = 0 THEN 'managar' ELSE 'Banker' END, -- Occupation typos  
 DATEADD(DAY, -(@i % 365), GETDATE()),  
 CASE WHEN @i % 11 = 0 THEN 'Closed' ELSE 'Active' END  
 );  
 SET @i = @i + 1;  
END

Repeat similar logic for Accounts and Transactions, injecting issues (nulls, out-of-range values, future dates, duplicates—via extra inserts with repeated keys, etc.).

**2. Use Third-Party Data Generation Tools**

You can use tools such as:

* SQL Server Data Generator (Redgate, Apex, dbForge etc.)
* Free online mock/synthetic data generators (export as CSV and bulk insert)
* Python or PowerShell scripts to create .sql or .csv files, then load them

**3. For Testing Purposes**

The short example helps you set up the logic, but to actually achieve 10,000+ rows, you need to use one of the above automation approaches.

**In summary:**

* The original insert statements are samples.
* For a full dataset, use looped inserts or a data generation tool/script to populate 10,000+ records **with the desired deliberate data issues** for cleaning and Power BI reporting.
* If you’d like a full sample T-SQL script that will generate 10,000 rows per table (or Python to do so), let me know your preferred method and I’ll provide it!

**WHAT ABOUT TRANSACTION AND ACCOUNT TABLES?**

Here are T-SQL scripts you can use to populate your **Accounts** and **Transactions** tables with 10,000 records each, deliberately introducing common "dirty data" issues for quality and cleaning practice. These scripts use WHILE loops, randomness, NULLs, and outliers.

**Generate 10,000 Accounts (with Data Quality Issues)**

-- Generate 10,000 Accounts with varied data and data quality issues  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Accounts (AccountID, CustomerID, AccountType, OpenDate, BranchID, Balance)  
 VALUES (  
 @i,  
 (@i % 10000) + 1, -- Randomly assign existing or non-existing CustomerID  
 CASE   
 WHEN @i % 10 = 0 THEN 'Savings Account'  
 WHEN @i % 10 = 1 THEN 'Saving'  
 WHEN @i % 10 = 2 THEN 'Current'  
 WHEN @i % 50 = 3 THEN 'Current Account'  
 ELSE 'Saving'   
 END, -- Inconsistent AccountType  
 DATEADD(DAY, -(@i % 3650), GETDATE()), -- Spread over 10 years  
 300 + (@i % 10), -- Various BranchIDs  
 CASE   
 WHEN @i % 30 = 0 THEN -500.00 -- Negative balances  
 WHEN @i % 40 = 0 THEN NULL -- NULL balance  
 ELSE CAST(ROUND(RAND() \* 20000, 2) AS DECIMAL(18,2))  
 END  
 );  
 SET @i = @i + 1;  
END

**Generate 10,000 Transactions (with Data Quality Issues)**

-- Generate 10,000 Transactions with varied data and data quality issues  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType, Description)  
 VALUES (  
 @i,  
 (@i % 10000) + 1, -- Random AccountID  
 DATEADD(DAY, -(@i % 365), GETDATE()), -- Dates spread over last year  
 CASE   
 WHEN @i % 15 = 0 THEN -1 \* (RAND() \* 1000) -- Negative outlier  
 WHEN @i % 20 = 0 THEN NULL -- NULL amount  
 ELSE CAST(ROUND((RAND() \* 5000), 2) AS DECIMAL(18, 2))  
 END,  
 CASE   
 WHEN @i % 4 = 0 THEN 'Deposit'  
 WHEN @i % 4 = 1 THEN 'Withdrawal'  
 WHEN @i % 4 = 2 THEN 'Fee'  
 ELSE 'Transfer'  
 END, -- Mixed casing/types  
 CASE   
 WHEN @i % 3 = 0 THEN CONCAT('Transaction ', @i)  
 ELSE NULL  
 END  
 );  
  
 -- Deliberately insert a duplicate TransactionID at i=100  
 IF @i = 100  
 BEGIN  
 INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType, Description)  
 VALUES (100, 200, GETDATE(), 500, 'Deposit', 'Duplicate Transaction');  
 END  
 SET @i = @i + 1;  
END

**How these scripts help:**

* They produce 10,000 rows each with realistic data, inconsistent formats, negative and NULL values, mislabels, duplicates, and future/past dates—all key issues for Power BI data cleaning exercises.
* You can adjust the logic for more/less messiness, more types, or to match your Customers table structure.

**Note:**  
Ensure your tables exist before running these scripts, and adjust the random data logic as needed to produce more business-relevant "dirty" data characteristics. For production-scale performance or further customization, consider using Python, SQL Server Data Generator, or ETL tools for more flexibility.

⁂

**TRY AGAIN**

Here's a revised and fully integrated T-SQL approach to programmatically generate approximately 10,000 records each for the Customers, Accounts, and Transactions tables, including the kinds of "dirty data" issues you want for Power BI practice. The logic introduces intentional errors and diverse formatting, consistent with your earlier requirements.

**1. Create the Tables**

CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 Name NVARCHAR(100),  
 Gender NVARCHAR(20),  
 DateOfBirth DATE,  
 Address NVARCHAR(200),  
 City NVARCHAR(100),  
 State NVARCHAR(50),  
 Zip NVARCHAR(20),  
 Email NVARCHAR(100),  
 Phone NVARCHAR(30),  
 Occupation NVARCHAR(100),  
 CreatedDate DATE,  
 Status NVARCHAR(20)  
);  
  
CREATE TABLE Accounts (  
 AccountID INT PRIMARY KEY,  
 CustomerID INT,  
 AccountType NVARCHAR(50),  
 OpenDate DATE,  
 BranchID INT,  
 Balance DECIMAL(18,2)  
);  
  
CREATE TABLE Transactions (  
 TransactionID INT PRIMARY KEY,  
 AccountID INT,  
 TransactionDate DATE,  
 Amount DECIMAL(18,2),  
 TransactionType NVARCHAR(30),  
 Description NVARCHAR(500)  
);

**2. Populate Customers with Dirty Data**

SET NOCOUNT ON;  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Customers (  
 CustomerID, Name, Gender, DateOfBirth, Address, City, State, Zip, Email, Phone, Occupation, CreatedDate, Status  
 )  
 VALUES (  
 @i,  
 CONCAT('User', CASE WHEN @i % 1000 = 0 THEN ' ' ELSE '' END, @i), -- occasional blank  
 CASE   
 WHEN @i % 9 = 0 THEN 'Male' -- inconsistent gender labels  
 WHEN @i % 9 = 1 THEN 'M'  
 WHEN @i % 9 = 2 THEN 'FEMALE'  
 WHEN @i % 9 = 3 THEN 'F'  
 WHEN @i % 9 = 4 THEN 'female'  
 ELSE NULL   
 END,  
 DATEADD(YEAR,-(20+(@i%30)),GETDATE()),  
 CONCAT(@i, CASE WHEN @i % 8 = 0 THEN '' ELSE ' Main St' END), -- missing address variant  
 CASE WHEN @i % 20 = 0 THEN 'bankar' ELSE 'New York' END, -- typos in city  
 'NY',  
 RIGHT('0000' + CAST(@i AS VARCHAR(4)),5),  
 CONCAT(  
 'user', @i,   
 CASE WHEN @i % 15 = 0 THEN '@EMAIL.com'  
 WHEN @i % 13 = 0 THEN ''  
 ELSE '@email.com'  
 END  
 ), -- inconsistent/missing email  
 CASE WHEN @i % 12 = 0 THEN NULL   
 WHEN @i % 12 = 1 THEN '555-123-4567'  
 WHEN @i % 12 = 2 THEN '(555) 123 4567'  
 ELSE CONCAT('555', @i)  
 END, -- phone format variations  
 CASE WHEN @i % 40 = 0 THEN 'managar'  
 WHEN @i % 15 = 0 THEN NULL  
 ELSE 'Banker'  
 END, -- occupation errors  
 DATEADD(DAY, -(@i % 3650), GETDATE()),  
 CASE WHEN @i % 16 = 0 THEN 'Closed' ELSE 'Active' END  
 );  
 -- Deliberate duplicate every 1000th entry  
 IF @i % 1000 = 0 INSERT INTO Customers SELECT \* FROM Customers WHERE CustomerID = @i;  
 SET @i = @i + 1;  
END

**3. Populate Accounts with Dirty Data**

SET NOCOUNT ON;  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Accounts (AccountID, CustomerID, AccountType, OpenDate, BranchID, Balance)  
 VALUES (  
 @i,  
 ((@i % 10000) + 1),  
 CASE   
 WHEN @i % 7 = 0 THEN 'Savings Account'  
 WHEN @i % 7 = 1 THEN 'Current'  
 WHEN @i % 10 = 0 THEN 'Saving'  
 ELSE 'Current Account'  
 END, -- account type variants  
 DATEADD(DAY, -(@i % 4000), GETDATE()),  
 300 + (@i % 15),  
 CASE WHEN @i % 21 = 0 THEN -500.00  
 WHEN @i % 17 = 0 THEN NULL  
 ELSE CAST(ROUND(RAND() \* 15000, 2) AS DECIMAL(18,2))  
 END  
 );  
 -- Deliberate duplicate every 1200th entry  
 IF @i % 1200 = 0 INSERT INTO Accounts SELECT \* FROM Accounts WHERE AccountID = @i;  
 SET @i = @i + 1;  
END

**4. Populate Transactions with Dirty Data**

SET NOCOUNT ON;  
DECLARE @i INT = 1;  
WHILE @i <= 10000  
BEGIN  
 INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType, Description)  
 VALUES (  
 @i,  
 ((@i % 10000) + 1),  
 CASE   
 WHEN @i % 2500 = 0 THEN DATEADD(DAY, (@i % 100), GETDATE()) -- occasional future date  
 ELSE DATEADD(DAY, -(@i % 900), GETDATE())  
 END,  
 CASE   
 WHEN @i % 15 = 0 THEN -1 \* (RAND() \* 1000)  
 WHEN @i % 20 = 0 THEN NULL  
 ELSE CAST(ROUND(RAND() \* 4000, 2) AS DECIMAL(18,2))  
 END,  
 CASE   
 WHEN @i % 5 = 0 THEN 'Deposit'  
 WHEN @i % 5 = 1 THEN 'Withdrawal'  
 WHEN @i % 5 = 2 THEN 'Fee'  
 WHEN @i % 5 = 3 THEN 'deposit' -- inconsistent case  
 ELSE 'Transfer'  
 END,  
 CASE   
 WHEN @i % 16 = 0 THEN NULL  
 ELSE CONCAT('Transaction ', @i)  
 END  
 );  
 -- Deliberate duplicate every 2000th entry  
 IF @i % 2000 = 0 INSERT INTO Transactions SELECT \* FROM Transactions WHERE TransactionID = @i;  
 SET @i = @i + 1;  
END

**How to Use:**

* Run each script after the tables are created.
* Adjust loop limits or error frequencies to tune your dataset size/complexity.
* You’ll have around 10,000 rows per table and ample data-quality challenges for Power BI analytics and cleaning!

**THESE ARE THE COLUMNS FROM TRANSACTIONS -**

TransactionID  
AccountID  
TransactionDate  
Amount  
TransactionType  
Description

THESE ARE THE COLUMNS FROM CUSTOMERS TABLE-  
CustomerID  
Name  
Gender  
DateOfBirth  
Address  
City  
State  
Zip  
Email  
Phone  
Occupation  
CreatedDate  
Status

THESE ARE THE COLUMNS FROM ACCOUNTS TABLE-  
COLUMN\_NAME  
AccountID  
CustomerID  
AccountType  
OpenDate  
BranchID  
Balance

i WANT TO COMBINE DATA FROM THESE TABLES AND I WANT ONLY A SINGLE TABLE TO BE CREATED THAT I WANT TO USE FOR POWER BI REPORT. SO WRITE SQL QEURY TO COMBINE THESE THREE TABLE

Certainly! To create a **single consolidated table** from your three source tables (Customers, Accounts, Transactions) for Power BI, you typically use JOIN operations.

Below is a SQL query to create a new table (let’s call it BankingCombined) that merges all relevant columns, **joining** Transactions to Accounts, and Accounts to Customers, so **each transaction contains all master/context details** of the customer and account involved.

**1. Create the New Combined Table**

CREATE TABLE BankingCombined AS  
SELECT  
 t.TransactionID,  
 t.AccountID,  
 t.TransactionDate,  
 t.Amount,  
 t.TransactionType,  
 t.Description,  
  
 a.AccountType,  
 a.OpenDate AS AccountOpenDate,  
 a.BranchID,  
 a.Balance AS AccountBalance,  
  
 c.CustomerID,  
 c.Name AS CustomerName,  
 c.Gender,  
 c.DateOfBirth,  
 c.Address,  
 c.City,  
 c.State,  
 c.Zip,  
 c.Email,  
 c.Phone,  
 c.Occupation,  
 c.CreatedDate AS CustomerCreatedDate,  
 c.Status AS CustomerStatus  
  
FROM  
 Transactions t  
 INNER JOIN Accounts a ON t.AccountID = a.AccountID  
 INNER JOIN Customers c ON a.CustomerID = c.CustomerID;

* This query will create a new table (BankingCombined) with all columns needed for reporting, with one row per transaction and all related account and customer context per transaction.

**Note:**

SELECT   
 t.TransactionID, t.AccountID, t.TransactionDate, t.Amount, t.TransactionType, t.Description,  
 a.AccountType, a.OpenDate AS AccountOpenDate, a.BranchID, a.Balance AS AccountBalance,  
 c.CustomerID, c.Name AS CustomerName, c.Gender, c.DateOfBirth, c.Address, c.City, c.State, c.Zip,  
 c.Email, c.Phone, c.Occupation, c.CreatedDate AS CustomerCreatedDate, c.Status AS CustomerStatus  
  
INTO BankingCombined  
  
FROM Transactions t  
 INNER JOIN Accounts a ON t.AccountID = a.AccountID  
 INNER JOIN Customers c ON a.CustomerID = c.CustomerID;

* The SELECT ... INTO syntax works in all supported SQL Server versions.

**2. Use the Combined Table in Power BI**

Now, simply connect Power BI to this BankingCombined table for reporting and data cleaning—**one flat table, no joins needed in Power BI**.

**3. (Optional) Customization**

* You can **add/remove** columns from the SELECT statement as needed.
* Use LEFT JOIN if you want to keep transactions even if account or customer is missing.

**Let me know if you want to include additional logic, perform filters, or add calculated columns as part of this query!**

**SELECT**

t.TransactionID, t.AccountID, t.TransactionDate, t.Amount, t.TransactionType, t.Description,  
a.AccountType, a.OpenDate AS AccountOpenDate, a.BranchID, a.Balance AS AccountBalance,  
c.CustomerID, c.Name AS CustomerName, c.Gender, c.DateOfBirth, c.Address, c.City, c.State, c.Zip,  
c.Email, c.Phone, c.Occupation, c.CreatedDate AS CustomerCreatedDate, c.Status AS CustomerStatus

INTO BankingCombined

FROM Transactions t  
LEFT JOIN Accounts a ON t.AccountID = a.AccountID  
LEFT JOIN Customers c ON a.CustomerID = c.CustomerID;

USE THE ABOVE QUERY AS REFERENCE AND RECOMMNED KPIS AND SUGGEST DAX MEASURES WHICH WILL BE REQUIRED

nOTE- WE WANT TO CREATE FOR 2 PAGE POWER BI REPORT

Absolutely! Based on your **BankingCombined** table (with all transaction, account, and customer context for reporting), here are **recommended KPIs and the corresponding DAX measures** for your two-page Power BI report.

**🌟 Recommended KPIs for BankingCombined Table**

**Page 1 — Deposit & Withdrawal Analysis / Customer Insights**

1. **Total Deposits**
   * **Description:** Sum of deposit transactions
   * **DAX:**

Total Deposits = CALCULATE(  
 SUM(BankingCombined.Amount),  
 BankingCombined.TransactionType = "Deposit"  
)

1. **Total Withdrawals**
   * **Description:** Sum of withdrawal transactions
   * **DAX:**

Total Withdrawals = CALCULATE(  
 SUM(BankingCombined.Amount),  
 BankingCombined.TransactionType = "Withdrawal"  
)

1. **Average Account Balance**
   * **Description:** Mean account balance at transaction time
   * **DAX:**

Average Account Balance = AVERAGE(BankingCombined.AccountBalance)

1. **Unique Active Customers**
   * **Description:** Number of unique customers with an "Active" status
   * **DAX:**

Active Customers = CALCULATE(  
 DISTINCTCOUNT(BankingCombined.CustomerID),  
 BankingCombined.CustomerStatus = "Active"  
)

1. **Transactions Volume Trend**
   * **Description:** Number of transactions over time (e.g., month)
   * **DAX:**

Transaction Volume = COUNTROWS(BankingCombined)

* + - Use this with a time-based visual (e.g., by TransactionDate).

**Page 2 — Product & Risk Metrics / Data Quality**

1. **Customer Churn Rate**
   * **Description:** Percentage of customers with status "Closed"
   * **DAX:**

Churn Rate =   
 DIVIDE(  
 CALCULATE(DISTINCTCOUNT(BankingCombined.CustomerID), BankingCombined.CustomerStatus = "Closed"),  
 CALCULATE(DISTINCTCOUNT(BankingCombined.CustomerID))  
 )

1. **Account Type Distribution**
   * **Description:** Number of accounts per account type
   * **DAX (used as Category Axis):**
     + Use a column or pie chart with **AccountType** as category, **COUNTROWS** or **DISTINCTCOUNT(AccountID)** for value.
2. **Negative Balance Accounts**
   * **Description:** Number of unique accounts with negative balance
   * **DAX:**

Negative Balance Accounts =   
 CALCULATE(  
 DISTINCTCOUNT(BankingCombined.AccountID),  
 BankingCombined.AccountBalance < 0  
 )

1. **Average Transaction Amount**
   * **Description:** Mean value of all transactions
   * **DAX:**

Average Transaction Amount = AVERAGE(BankingCombined.Amount)

1. **Data Quality: Null Transaction Descriptions**
   * **Description:** Percentage or count of transactions with missing/null Description
   * **DAX:**

Null Descriptions =   
 CALCULATE(  
 COUNTROWS(BankingCombined),  
 ISBLANK(BankingCombined.Description)  
 )

* + *Turn this into a % by dividing by total transactions count.*

**Suggested Visual Layout**

**Page 1: Customer & Transaction Overview**

* **Cards:** Total Deposits, Total Withdrawals, Average Account Balance, Active Customers
* **Line/Column Chart:** Transaction Volume Trend by TransactionDate (Month)
* **Table/Matrix:** Top customers by deposit amount

**Page 2: Product, Risk & Data Quality**

* **Donut/Pie Chart:** Account Type Distribution
* **Card:** Churn Rate, Negative Balance Accounts
* **Gauge:** Average Transaction Amount
* **Bar Chart:** Number of Closed vs. Active Customers
* **Card or Table:** Null Transaction Descriptions (Data Quality)

**Extra DAX Tips:**

* **For filters** like "last month," add CALCULATE with DATEADD or FILTER.
* **For advanced data quality**: You can also count duplicates using DAX if helpful (let me know if you want those formulas).

**You can build all these measures and visuals directly from the BankingCombined table in Power BI.**  
If you want tailored DAX for a specific metric or would like sample report layouts, just ask!

**PLEASE RECOMMNED kpis**

Here are **recommended KPIs** you can track using your consolidated BankingCombined table, tailored for a two-page Power BI report. These cover deposits, withdrawals, balances, customer engagement, product mix, risk, and data quality:

**Page 1: Customer & Transaction Overview KPIs**

1. **Total Deposits**
   * Sum of all deposit transactions.
2. **Total Withdrawals**
   * Sum of all withdrawal transactions.
3. **Average Account Balance**
   * Mean balance value across accounts.
4. **Number of Active Customers**
   * Unique count of customers with status "Active."
5. **Transaction Volume Trend**
   * Count of all transactions (can be trended by month or date).

**Page 2: Product, Risk & Data Quality KPIs**

1. **Customer Churn Rate**
   * Percentage of customers with status "Closed" versus total.
2. **Account Type Distribution**
   * Number or percentage of accounts per account type.
3. **Negative Balance Accounts**
   * Count of accounts with negative balances.
4. **Average Transaction Amount**
   * Mean transaction size across all transaction types.
5. **Null or Missing Transaction Descriptions**
   * Count or percentage of transactions with blank/missing descriptions (to illustrate data quality issues).

**KPI Visual Suggestions**

* **Cards:** Headline metrics (deposits, withdrawals, balances, active customers, churn rate)
* **Column/Pie/Donut:** Product and account type distribution, closed vs. active customers
* **Line/Column Charts:** Trends (transaction volume over time)
* **Gauge:** Ratios (e.g., churn, negative balance rate)
* **Bar/Table:** Data quality indicators (nulls, duplicates, errors)

If you need DAX formulas for these KPIs or want sample visual arrangements, let me know!

**RECOMMNED IN A WAY THAT INCLUDES IT'S VISUALIXATION AND DAX**

Here are **recommended KPIs** for your consolidated BankingCombined table, each paired with an ideal Power BI visualization and a sample DAX formula to create powerful, actionable metrics. This is structured for a two-page Power BI report.

**Page 1: Customer & Transaction Overview**

|  |  |  |
| --- | --- | --- |
| KPI Name | Visualization | DAX Formula Example |
| **Total Deposits** | Card, Line/Area Trend Chart | Total Deposits = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType] = "Deposit") |
| **Total Withdrawals** | Card, Line/Area Trend Chart | Total Withdrawals = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType] = "Withdrawal") |
| **Average Account Balance** | Card, Gauge | Average Account Balance = AVERAGE(BankingCombined[AccountBalance]) |
| **Active Customers** | Card, Bar Chart | Active Customers = CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]), BankingCombined[CustomerStatus] = "Active") |
| **Transaction Volume Trend** | Line/Column Chart by Date | Transaction Volume = COUNTROWS(BankingCombined) (use TransactionDate on X-axis) |

**Page 2: Product, Risk & Data Quality**

|  |  |  |
| --- | --- | --- |
| KPI Name | Visualization | DAX Formula Example |
| **Customer Churn Rate** | Card, Donut/Pie | Churn Rate = DIVIDE(CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]), BankingCombined[CustomerStatus] = "Closed"), CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]))) |
| **Account Type Distribution** | Bar, Donut/Pie Chart | Use AccountType as category, DISTINCTCOUNT(BankingCombined[AccountID])/COUNTROWS for values |
| **Negative Balance Accounts** | Card, Gauge, Bar | Negative Balance Accounts = CALCULATE(DISTINCTCOUNT(BankingCombined[AccountID]), BankingCombined[AccountBalance] < 0) |
| **Average Transaction Amount** | Card, Gauge | Average Transaction Amount = AVERAGE(BankingCombined[Amount]) |
| **Transactions with Null Description** | Card, Bar Chart, Table | Null Desc Transactions = CALCULATE(COUNTROWS(BankingCombined), ISBLANK(BankingCombined[Description])) |

**How to Use These in Power BI**

* **Cards** for single values (totals, averages, active count, churn).
* **Column/Line Charts** for time trends (volume, deposits, withdrawals by month/week).
* **Pie/Donut Charts** for distribution (account type, active vs. closed customers).
* **Gauges** for progress-style KPIs (ratios, negative balances).
* **Bar/Table Visuals** for deeper breakdown (data quality, product/risk segmentation).

Let me know if you need formatted DAX ready to copy, complex time intelligence (like rolling averages), or sample report wireframes for Power BI!

**ADD DIFFERENT TYPE OF KPI VISUALS AS YOU HAVE SUGGESTED MOSTLY CARD**

Here’s a revised set of **KPIs**, each with a **different Power BI visualization type** (not just Cards), using your consolidated BankingCombined table and paired with example DAX formulas.

**Page 1: Customer & Transaction Overview**

|  |  |  |
| --- | --- | --- |
| KPI Name | Recommended Visualization | Example DAX Formula |
| **Total Deposits** | **Line/Area Chart** (trend over time by TransactionDate) | Total Deposits = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType]="Deposit") |
| **Total Withdrawals** | **Clustered Column Chart** (trend over time or by AccountType) | Total Withdrawals = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType]="Withdrawal") |
| **Active Customers** | **Stacked Bar Chart** (by City, Status as legend) | Active Customers = CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]), BankingCombined[CustomerStatus]="Active") |
| **Transaction Volume Trend** | **Line Chart** (by TransactionDate) | Transaction Volume = COUNTROWS(BankingCombined) |
| **Average Account Balance** | **Gauge** | Average Account Balance = AVERAGE(BankingCombined[AccountBalance]) |

**Page 2: Product, Risk & Data Quality**

|  |  |  |
| --- | --- | --- |
| KPI Name | Recommended Visualization | Example DAX Formula |
| **Customer Churn Rate** | **Donut Chart** | Churn Rate = DIVIDE(CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]), BankingCombined[CustomerStatus]="Closed"), CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]))) |
| **Account Type Distribution** | **Pie Chart** (AccountType) | Use AccountType as legend, DISTINCTCOUNT(BankingCombined[AccountID]) as value |
| **Negative Balance Accounts** | **Gauge** | Negative Balance Accounts = CALCULATE(DISTINCTCOUNT(BankingCombined[AccountID]), BankingCombined[AccountBalance]<0) |
| **Average Transaction Amount** | **Clustered Bar Chart** (by TransactionType) | Average Transaction Amount = AVERAGE(BankingCombined[Amount]) |
| **Transactions with Null Description** | **Table or Stacked Column Chart** (by AccountType, count of null descriptions) | Null Desc Transactions = CALCULATE(COUNTROWS(BankingCombined), ISBLANK(BankingCombined[Description])) |

**How to Use These Visuals**

* **Trends**: Place line or area charts on primary report pages to show volume or amount changes by date.
* **Compositions/Segmentation**: Use pie/donut/bar charts for breakdowns (account type, churn).
* **Risks/Thresholds**: Gauges show proportions approaching critical values (averages, negatives).
* **Data Quality**: Show tables or bar charts for count of records with missing/dirty data fields.

These visuals will keep both pages of your Power BI report visually compelling, interactive, and full of actionable banking metrics. Let me know if you need wireframe mockups or more advanced visual/DAX customizations!

**i want this kpis on first page**

Here’s a recommended set of **KPIs for Page 1** of your Power BI report, each with a visually distinct, non-card visualization and corresponding DAX formulas—optimized for your consolidated BankingCombined table:

**First Page KPIs, Visual Types, and DAX Measures**

|  |  |  |
| --- | --- | --- |
| KPI Name | Visualization Type | Example DAX Formula |
| **Total Deposits (Trend)** | **Line Chart** by TransactionDate | Total Deposits = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType]="Deposit") |
| **Total Withdrawals (By Account Type)** | **Clustered Column Chart** (AccountType axis) | Total Withdrawals = CALCULATE(SUM(BankingCombined[Amount]), BankingCombined[TransactionType]="Withdrawal") |
| **Active Customers by City** | **Stacked Bar Chart** (City axis, Status as Legend) | Active Customers = CALCULATE(DISTINCTCOUNT(BankingCombined[CustomerID]), BankingCombined[CustomerStatus]="Active") |
| **Transaction Volume Over Time** | **Line Chart** (by TransactionDate) | Transaction Volume = COUNTROWS(BankingCombined) |
| **Average Account Balance** | **Gauge** | Average Account Balance = AVERAGE(BankingCombined[AccountBalance]) |

**How To Design the Page**

* **Visual Diversity**: Mix line, column, bar, and gauge visuals—don’t repeat card visuals.
* **Interactivity**: Provide slicers (filters) for date, account type, or city for richer exploration.
* **Insights**: These visuals let you analyze trends, product mix, customer engagement, and financial health at a glance.

You can copy these DAX formulas into Power BI to create the measures and select the suggested visuals during report design.

If you want additional KPIs or mockup sketches for the next page, just ask!

⁂

**A**

D

3

Transactions by Type

Monthly Transaction Amount

Top N Customers by Transaction Value

Average Account Balance

Total Balance by Account Type

Inactive Accounts (last 90 days)

Description

Volume or pr

Aggregate balances per account type

Accounts with no transactions within the past 90 days

Total transaction value grouped by month

Customers with the highest transaction value

Mean balance per account

tions by type (Credit/Debit)

Visual

Pie Chart/Stacked Colun

Line or Area Chart

Top N Bar Chart

Column Chart/Gauge

Clustered Bar/Column

Clustered Bar

Here’s how you can structure your **Power BI Page 1 KPIs** (based on your provided intent and matching them to type, description, visual, and suggested DAX for your BankingCombined table):

|  |  |  |  |
| --- | --- | --- | --- |
| KPI Name | Description | Visual Type | Example DAX Formula |
| **Transactions by Type** | Volume or proportion by credit/debit | Pie Chart / Stacked Column | Transaction Volume by Type = COUNTROWS(BankingCombined)  Use "TransactionType" as legend/axis. |
| **Monthly Transaction Amount** | Total transaction value grouped by month | Line or Area Chart | Monthly Transaction Amount = SUM(BankingCombined[Amount])  Plot by "Month" (from TransactionDate). |
| **Top N Customers by Transaction Value** | Customers with the highest transaction value | Top N Bar Chart | Customer Transaction Value = SUM(BankingCombined[Amount])  Rank by Customer, filter top N. |
| **Average Account Balance** | Mean balance per account | Column Chart or Gauge | Average Account Balance = AVERAGE(BankingCombined[AccountBalance]) |
| **Total Balance by Account Type** | Aggregate balances per account type | Clustered Bar/Column | Total Balance by Account Type = SUM(BankingCombined[AccountBalance])  Grouped by AccountType. |
| **Inactive Accounts (last 90 days)** | Accounts with no transactions in past 90 days | Clustered Bar | Inactive Accounts = CALCULATE(DISTINCTCOUNT(BankingCombined[AccountID]), NOT BankingCombined[TransactionDate] >= TODAY()-90) |

**How to Build These KPIs in Power BI:**

* **Pie or Stacked Column**: Show proportion of each TransactionType (e.g., Credit vs. Debit).
* **Line/Area Chart**: Plot SUM(Amount) with TransactionDate (month/year) on the X-axis.
* **Top N Bar Chart**: Use a bar chart, rank Customers by sum of transaction values, display top N.
* **Column Chart/Gauge**: Display the average account balance, use a gauge for a target/benchmark view.
* **Clustered Bar/Column**: Show total balances grouped by account type (e.g., Savings vs. Current).
* **Clustered Bar (Inactive Accounts)**: Highlight accounts with no activity in the last 90 days—group by AccountType/Branch.

You can implement these KPIs with their visuals and DAX directly in Power BI using your BankingCombined table, supporting clear and insightful reporting. Let me know if you need fully written DAX or want example visuals!

⁂

**the dax formula looks incorrect please recheck**

You're right—some of the earlier DAX expressions (especially for segmenting/filtering) were not using proper DAX syntax. Here are **corrected DAX formulas** for each KPI, specifically tailored for the columns in your BankingCombined table.

**1. Transactions by Type**

**Pie Chart / Stacked Column**

Transactions by Type = COUNTROWS(BankingCombined)

* For chart: Use TransactionType as the Legend/Axis.  
  (The count will automatically group by type in the visual.)

**2. Monthly Transaction Amount**

**Line or Area Chart**

Monthly Transaction Amount =   
SUM(BankingCombined[Amount])

* Place TransactionDate (or Month portion extracted using Power BI's automatic date table or MONTH(BankingCombined[TransactionDate])) on X-axis for monthly trend.

**3. Top N Customers by Transaction Value**

**Top N Bar Chart**

Total Transaction Value by Customer =   
CALCULATE(  
 SUM(BankingCombined[Amount])  
)

* Visual: Use CustomerName or CustomerID as Axis, the above measure as Value. Use "Top N" filter in the visual to restrict to top customers.

**4. Average Account Balance**

**Column Chart or Gauge**

Average Account Balance =   
AVERAGE(BankingCombined[AccountBalance])

**5. Total Balance by Account Type**

**Clustered Bar/Column**

Total Balance by Account Type =   
CALCULATE(  
 SUM(BankingCombined[AccountBalance])  
)

* Use AccountType as axis in the visual.

**6. Inactive Accounts (last 90 days)**

**Clustered Bar**

To count distinct accounts that have **no transactions in the last 90 days**:

Inactive Accounts (last 90 days) =   
VAR RecentAccounts =  
 CALCULATETABLE (  
 VALUES(BankingCombined[AccountID]),  
 BankingCombined[TransactionDate] >= TODAY() - 90  
 )  
RETURN  
CALCULATE(  
 DISTINCTCOUNT(BankingCombined[AccountID]),  
 NOT (BankingCombined[AccountID] IN RecentAccounts)  
)

Or, simpler (if needed as a label column for filtering visuals):

IsInactive90 =  
IF (  
 CALCULATE (  
 COUNTROWS(BankingCombined),  
 FILTER(  
 BankingCombined,  
 BankingCombined[AccountID]=EARLIER(BankingCombined[AccountID]) &&  
 BankingCombined[TransactionDate]>=TODAY()-90  
 )  
 )=0,  
 "Inactive",  
 "Active"  
)

Then in the visual, use IsInactive90="Inactive" for filtering/bar breakdown.

**Tip:**  
For each DAX, use appropriate columns in the Visual’s Fields pane to ensure correct grouping (e.g., for monthly trends, top N, or by account type).

If you want me to tailor these DAX formulas even further (for example, for exact date handling or Power BI time intelligence), please specify, and I’ll provide advanced expressions!

**suggest me colour code for my pie chart and theme for my report**

Here are **color palette and theme suggestions** for your Power BI pie chart and overall report, designed for banking/business analytics—focusing on clarity, contrast, and professionalism.

**Pie Chart Color Codes**

Choose 5–7 distinguishable, accessible colors for categories like TransactionType or AccountType. Here are widely accepted hex codes (color names in parentheses):

* #2E86AB (Blue)
* #F6C85F (Gold)
* #6FB07F (Green)
* #FF6F69 (Coral Red)
* #9B59B6 (Purple)
* #95A5A6 (Gray)
* #F79E02 (Orange)

These colors offer strong visual separation on light or dark backgrounds, meet accessibility contrast guidelines, and avoid confusion for colorblind users.

**Report Theme Recommendation**

**Modern Business/Finance Theme:**

* **Background:** Pure White #FFFFFF or very light gray #F7F9FB
* **Accent 1 (Headers):** Deep Blue #2E4053
* **Accent 2 (Highlight):** Brand Green #27AE60 or Gold #F6C85F
* **Text:** Nearly Black #1A1A1A
* **Borders/Lines:** Soft Gray #D5DBDB
* **Cards/Visual backgrounds:** Slightly off-white #FAFAFA or subtle blue tint #EFF5FB

**Use Gradient or Solid fills** for visuals, sparingly add shadow on cards for a professional, fresh look.

**How to Apply in Power BI**

* In Power BI Desktop:  
  Go to **View > Themes > Customize current theme > Colors**, and input these hex codes.
* Assign pie segments manually if needed for brand consistency and color meaning.

**Tip:**  
If your bank or organization already has brand colors, adjust above hex codes to match, ensuring contrast and visual accessibility stay strong.

Let me know if you want a ready-made Power BI JSON theme file or custom branding!

**Inactive Accounts (last 90 days)**

Accounts with no transactions in past 90 days  
Clustered Bar  
Inactive Accounts = CALCULATE(DISTINCTCOUNT(BankingCombined[AccountID]), NOT BankingCombined[TransactionDate] >= TODAY()-90)

but we already have customer status column so use that directly

Certainly! Since your **BankingCombined** table already includes a Customer Status column (e.g., "Active," "Closed," etc.), you can directly use that status—without calculating recency from the transaction date—to define inactive accounts.

**DAX for Inactive Accounts Using CustomerStatus**

If “Inactive” or “Closed” customers are to be counted:

Inactive Accounts =   
CALCULATE(  
 DISTINCTCOUNT(BankingCombined[AccountID]),  
 BankingCombined[CustomerStatus] = "Closed"  
 // or use "Inactive" according to your data semantics  
)

* If your inactive status is labeled differently (such as "Inactive" or any custom string), replace "Closed" above with the correct status value.

**How to Use in Power BI**

* Use this measure in a **clustered bar chart** to display inactive accounts by city, account type, or branch.
* Optionally, you can break down further by AccountType or City for more detailed reporting.

**Summary:**  
Using your CustomerStatus column is the most efficient and business-aligned way to identify inactive accounts, as it reflects the actual account/customer lifecycle directly—no need to infer inactivity from transaction date if status is reliable.

**let's add some slicers**

Adding slicers to your Power BI report allows users to interactively filter your KPIs and visuals, offering deeper insights and tailored analytics. Here are intelligent slicer recommendations for your BankingCombined table based on best practices for banking analytics:

**Recommended Slicers to Add**

1. **Transaction Date**
   * **Type:** Date range or relative date slicer
   * **Purpose:** Filter all visuals by a selected period (e.g., year, month, last 30 days, quarter).
2. **Account Type**
   * **Type:** Dropdown or list
   * **Purpose:** See metrics for specific product types (e.g., Savings, Current).
3. **Customer Status**
   * **Type:** Dropdown (Active, Closed, etc.)
   * **Purpose:** Instantly split KPIs or visuals by customer lifecycle.
4. **City or State**
   * **Type:** Dropdown or search
   * **Purpose:** Geo-segment KPIs (e.g., churn, deposits, average balance) for regional management.
5. **Transaction Type**
   * **Type:** Dropdown or list (Deposit, Withdrawal, Fee, etc.)
   * **Purpose:** Filter visuals/KPIs by transaction type for focused analysis.
6. **Branch ID**
   * **Type:** Dropdown
   * **Purpose:** Branch-level comparison and operational insights.
7. **Account Status** (if available)
   * **Type:** Dropdown
   * **Purpose:** Further refine by open/closed/inactive accounts.

**Advanced Slicer Tips**

* You can synchronize date or region slicers across both report pages for consistent filtering.
* For Top N visuals (like Top Customers), use additional numerical slider slicers (“Top N”) for user-defined ranking.
* Slicers can be moved to a side panel or top ribbon for a clean, accessible layout.

**How to Add Slicers in Power BI:**

* In the Power BI Desktop “Visualizations” pane, select the **Slicer** visual and drag your desired field (e.g., TransactionDate, City, etc.) to its “Field” box.
* Customize slicer interaction—calendar, dropdown, list, or search bar—as fits your report’s user experience.

**Properly chosen slicers ensure your KPIs remain dynamic and relevant, empowering end-users to explore your banking data on their terms. Let me know if you need specific slicer formatting tips or want to implement cross-filtering between pages!**

**I want a suitable title for my dashboard saying something like it is an end to end tutorial on how to use AI (perpexility AI) for everything.**

then i am goi=n to post it on linked in and i will put down all the steps that i have used in the post from prompts to everything

Here are some **engaging dashboard and LinkedIn post title ideas** that capture your end-to-end, AI-powered, Power BI banking analytics journey using Perplexity AI, emphasizing its tutorial and automation aspects:

**Dashboard Title Suggestions**

* **"End-to-End Banking Analytics: An Interactive Tutorial Using Perplexity AI & Power BI"**
* **"From Data Generation to Insights: Leveraging Perplexity AI for Automated Banking Analytics"**
* **"Smart Banking Dashboard: A Comprehensive Walkthrough With Perplexity AI and Power BI"**
* **"Transforming Raw Data to Business Intelligence: An AI-Powered Banking Analytics Journey"**
* **"AI-Driven Data Story: Building a Banking Dashboard From Scratch With Perplexity AI"**

**LinkedIn Post Headline Ideas**

* **"How I Used Perplexity AI to Build an End-to-End Banking Analytics Solution in Power BI: Full Tutorial & Prompts"**
* **"AI + Analytics: Step-By-Step Guide to Creating a Power BI Banking Dashboard Using Perplexity AI"**
* **"A Complete Walkthrough—Automating Data Prep, Cleaning, and Visualization With Perplexity AI"**

**Tips for Your LinkedIn Post:**

* Lead with your top-level title (“End-to-End Banking Analytics: An Interactive Tutorial Using Perplexity AI & Power BI”).
* In your introductory sentence, clarify that you’re sharing all steps, prompts, DAX, SQL, visuals, and insights.
* Example intro:

"In this post, I’ll show you every step I followed to build a fully automated banking analytics dashboard—from dirty data creation to beautiful Power BI reports—using Perplexity AI as my assistant. You’ll get all prompts, scripts, DAX, and visuals so you can replicate this yourself!"

This framing will attract both data professionals and those interested in the practical, real-world use of AI in analytics and business intelligence. If you want a LinkedIn snippet or summary paragraph, let me know!