The Bank Database Case

James William Martinsson Co

This document outlines the functionalities, structure and rationale behind the development of this banking database. The primary objective of this project was to design a *conceptual*, *logical* and *physical* database model tailored for banking operations.

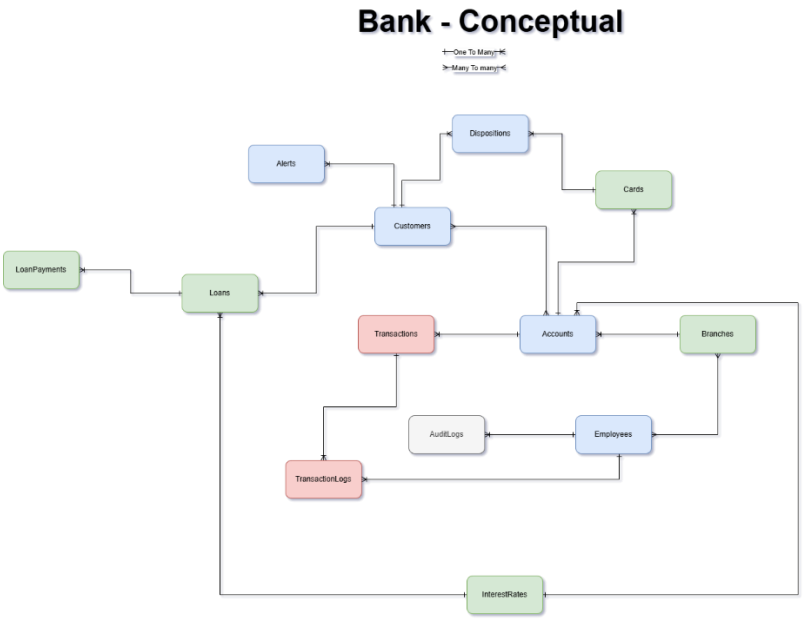
The *conceptual* and *logical* models were created using **Draw.io**, while the *physical* implementation was developed using **SQL Server Management Studio (SSMS).**

This project represents the final individual assignment for the "***Database Design and Modelling***" course at Nackademin.

While some functionalities may require integration with external systems, the database has been designed to be adaptable to the requirements of most modern banks.

Table of Contents

Note: Shortcut! You can click on one of the titles in the Table of Contents if you want to jump to those sections.



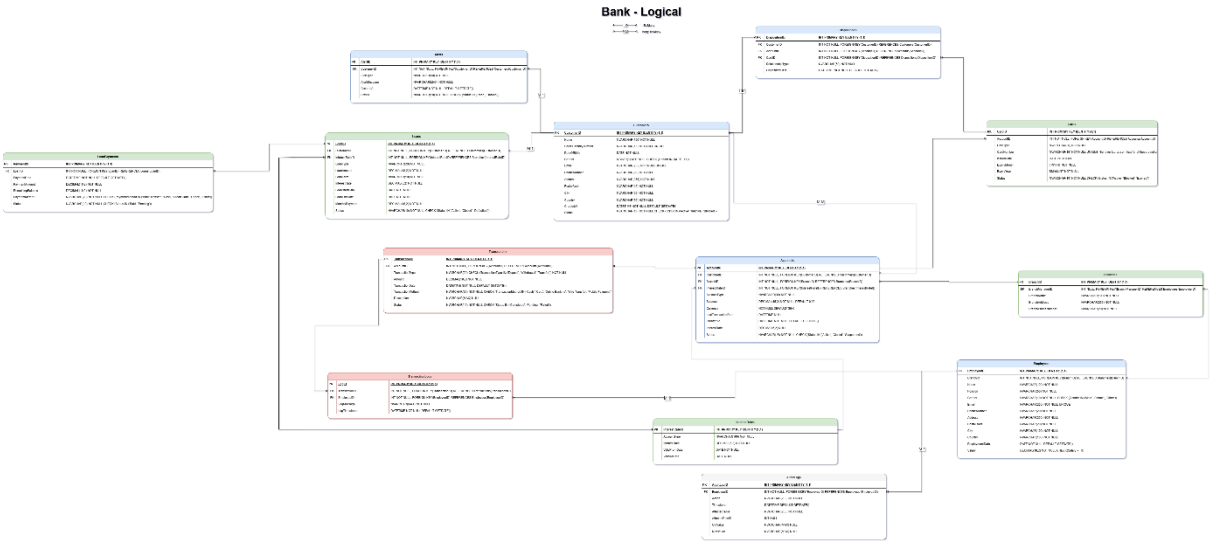
(The original file is provided separately.)

Conceptual Model

For us to create a conceptual model, we needed to do some research. I decided to do that with the help of my current bank “Handelsbanken”. Having access to a bank as a customer helps me think of ideas and what kind of tables I need to create for a functional and well-structured database model for all kinds of bank companies.

The tables:

- **Accounts** = Stores details of each bank account.
- **Alerts** = Stores notifications and alerts for customers (ex. low balance, large withdrawal etc.)
- **AuditLogs** = Stores logs of important system actions for auditing purposes.
- **Branches** = Stores information about the bank branches.
- **Cards** = Stores information about credit and debit cards linked to accounts.
- **Customers** = Stores personal information of customers.
- **Dispositions** = Links customers to cards and accounts, representing relationships.
- **Employees** = Stores information about employees.
- **InterestRates** = Stores information about the interest rates for different account types.
- **LoanPayments** = Tracks payment made towards loan
- **Loans** = Stores details about loans taken by customers.
- **Transactions** = Stores all the transactions made on bank accounts.
- **TransactionLogs** = Tracks transaction activities such as errors and tracking fraud attempts.



(The original file is provided separately.)

Logical Model

After the conceptual model was finished, it was then time to go further in the project and create and finalize our logical model. We've created a model that shows detailed tables with columns and datatypes. As well as using Crow's foot notation to indicate the different relations of the tables. With this, we can create our SQL Script and finalize our project. We also made sure to follow the 3NF for our model.

Tables and their one-to-many / many-to-many relationships with other tables:

Table	One-To-Many (1:M)	Many-To-Many (M:M)
Accounts	Branches, Cards, InterestRates, Transactions	Customers
Alerts	Customers	
AuditLogs	Employees	
Branches	Accounts	Employes
Cards	Accounts, Dispositions	
Customers	Alerts, Dispositions, Loans	Accounts
Dispositions	Cards, Customers	Cards ←→ Dispositions ← → Customers
Employees	AuditLogs, TransactionLogs	Branches
InterestRates	Accounts, Loans	
Loans	Customers, InterestRates, LoanPayments	
Transactions	Accounts, TransactionLogs	
TransactionLogs	Transactions, Employees	

SQL Script

Creation of the Database and Schemas

```sql
/*_____ Database _____*/

USE master; -- Makes sure that there's no other database selected and so that we can execute the code and create a new database.

IF EXISTS(SELECT * FROM sys.databases WHERE name = 'Bank') -- Drops the database if it exists.
    BEGIN
        ALTER DATABASE Bank SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
        DROP DATABASE Bank
    END
GO

IF NOT EXISTS(SELECT * FROM sys.databases WHERE name = 'Bank') -- Creates the database if it doesn't exist.
    BEGIN
        CREATE DATABASE [Bank]
    END
GO

/*_____ Schema _____*/

USE Bank; -- Makes sure to choose the new database we just created
GO

SET NOCOUNT ON;

IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'Users') -- Ensures that if the Schema doesn't exist, then it'll create the schemas, in this case 'Users' or 'Logs' and etc.
    EXEC('CREATE SCHEMA Users;');
GO

IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'Security')
    EXEC('CREATE SCHEMA Security;');
GO

IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'Bank')
    EXEC('CREATE SCHEMA Bank;');
GO

IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'Finance')
    EXEC('CREATE SCHEMA Finance;');
GO
```

These codes ensure that any database or schema with the same name is removed and then the database and schemas can then be created successfully without any errors.

Creation of the Tables

```sql
/*_____ Tables _____*/

CREATE TABLE Users.Customers (
    CustomerID INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL,
    SocialSecurityNumber NVARCHAR(13) NOT NULL UNIQUE,
    DateOfBirth DATE NOT NULL,
    Gender NCHAR(1) NOT NULL CHECK (Gender IN ('M', 'F', 'O')),
    Email NVARCHAR(255) NOT NULL UNIQUE,
    PhoneNumber NVARCHAR(20) NOT NULL,
    Address NVARCHAR(255) NOT NULL,
    PostalCode NVARCHAR(10) NOT NULL,
    City NVARCHAR(50) NOT NULL,
    Country NVARCHAR(60) NOT NULL,
    CreatedAt DATETIME NOT NULL DEFAULT GETDATE(),
    Status NVARCHAR(10) NOT NULL CHECK (Status IN ('Active', 'Inactive', 'Blocked'))
);

CREATE TABLE Users.Alerts (
    AlertID INT PRIMARY KEY IDENTITY(1,1),
    CustomerID INT NOT NULL,
    AlertType NVARCHAR(50) NOT NULL,
    AlertMessage NVARCHAR(MAX) NOT NULL,
    CreatedAt DATETIME NOT NULL DEFAULT GETDATE(),
    Status NVARCHAR(10) NOT NULL CHECK (Status IN ('Read', 'Unread')),
    CONSTRAINT FK_Alerts_Customers FOREIGN KEY (CustomerID) REFERENCES Users.Customers(CustomerID) ON DELETE CASCADE
);

CREATE TABLE Bank.Branches (
    BranchID INT PRIMARY KEY IDENTITY(1,1),
    BranchManagerID INT NULL,
    BranchName NVARCHAR(100) NOT NULL,
    BranchAddress NVARCHAR(255) NOT NULL,
    BranchPhoneNumber NVARCHAR(20) NOT NULL
);

CREATE TABLE Users.Employees (
    EmployeeID INT PRIMARY KEY IDENTITY(1,1),
    BranchID INT NOT NULL,
    Name NVARCHAR(100) NOT NULL,
    Position NVARCHAR(50) NOT NULL,
    Gender NCHAR(1) NOT NULL CHECK (Gender IN ('M', 'F', 'O')),
    Email NVARCHAR(255) NOT NULL UNIQUE,
    PhoneNumber NVARCHAR(20) NOT NULL,
    Address NVARCHAR(255) NOT NULL,
    PostalCode NVARCHAR(10) NOT NULL,
    City NVARCHAR(100) NOT NULL,
    Country NVARCHAR(100) NOT NULL,
    EmploymentDate DATE NOT NULL DEFAULT GETDATE(),
    Salary DECIMAL(18,2) NOT NULL CHECK(Salary >= 0),
    CONSTRAINT FK_Employees_Branches FOREIGN KEY (BranchID) REFERENCES Bank.Branches(BranchID)
);

ALTER TABLE Bank.Branches -- Has to be added after Branches and Employees tables are created so that we can create these two tables.
    ADD CONSTRAINT FK_Branches_Employees FOREIGN KEY (BranchManagerID) REFERENCES Users.Employees(EmployeeID);

CREATE TABLE Bank.InterestRates (
    InterestRateID INT PRIMARY KEY IDENTITY(1,1),
    AccountType NVARCHAR(100) NOT NULL,
    InterestRate DECIMAL(5,2) NOT NULL,
    ValidFromDate DATE NOT NULL,
    ValidToDate DATE NULL
    );
```

We created the tables and ensured that the tables are in the right orders to prevent errors with foreign keys. (We created Bank.Branches without the foreign key from Employees and added it afterwards with the Alter code as seen in the screenshot above.)

Insertion of Fake Data

```
/*_____ Data Insertion _____*/

INSERT INTO Users.Customers
    (Name, SocialSecurityNumber, DateOfBirth, Gender, Email, PhoneNumber, Address, PostalCode, City, Country, Status)
VALUES
    ('Johan Svensson', '19880415-7845', '1988-04-15', 'M', 'johan.svensson@gmail.com', '+46738911234', 'Storgatan 12', '114 56', 'Stockholm', 'Sweden', 'Active'),
    ('Fatima Hussein', '19950610-4521', '1995-06-10', 'F', 'fatima.hussein@yahoo.com', '+46763224567', 'Bergsgatan 23', '211 12', 'Malmö', 'Iraq', 'Active'),
    ('Anders Karlsson', '20010821-9856', '2001-08-21', 'M', 'anders.karlsson@hotmail.com', '+46782347654', 'Linnégatan 45', '413 04', 'Göteborg', 'Sweden', 'Inactive'),
    ('Sofia Zhang', '20050411-5673', '2005-04-11', 'F', 'sofia.zhang@gmail.com', '+46765438912', 'Kungsgatan 8', '753 20', 'Uppsala', 'China', 'Active'),
    ('Oskar Berg', '19991202-3456', '1999-12-02', 'M', 'oskar.berg@gmail.com', '+46734598761', 'Drottninggatan 30', '803 10', 'Gävle', 'Sweden', 'Blocked'),
    ('Lena Persson', '19830527-1234', '1983-05-27', 'F', 'lena.persson@gmail.com', '+46768765432', 'Huvudvägen 5', '541 45', 'Skövde', 'Sweden', 'Active'),
    ('Ahmed Ali', '20020814-9872', '2002-08-14', 'M', 'ahmed.ali@gmail.com', '+46769876543', 'Rosengårdsvägen 6', '213 66', 'Malmö', 'Somalia', 'Inactive'),
    ('Emma Lindström', '19970303-7654', '1997-03-03', 'F', 'emma.lindstrom@gmail.com', '+46765412389', 'Torsgatan 18', '411 03', 'Göteborg', 'Sweden', 'Blocked'),
    ('David Brown', '19900129-4321', '1990-01-29', 'M', 'david.brown@gmail.com', '+46768901234', 'Sveavägen 60', '113 59', 'Stockholm', 'USA', 'Active'),
    ('Aliyah Mohamed', '20070930-6543', '2007-09-30', 'F', 'aliyah.mohamed@gmail.com', '+46769871234', 'Lantmannagatan 9', '214 50', 'Malmö', 'Eritrea', 'Inactive'),
    ('Elliot Larsson', '20100205-8765', '2010-02-05', 'O', 'elliot.larsson@gmail.com', '+46769876521', 'Vasagatan 3', '722 11', 'Västerås', 'Sweden', 'Active'),
    ('Isabella Rossi', '19920418-4323', '1992-04-18', 'F', 'isabella.rossi@gmail.com', '+46768904321', 'Östra Hamngatan 4', '411 10', 'Göteborg', 'Italy', 'Blocked'),
    ('Kristian Nilsson', '20060317-5432', '2006-03-17', 'M', 'kristian.nilsson@gmail.com', '+46765498721', 'Brunnsgatan 19', '111 38', 'Stockholm', 'Sweden', 'Inactive'),
    ('Morgan Sjöberg', '19891224-3214', '1989-12-24', 'O', 'morgan.sjoberg@gmail.com', '+46769871239', 'Björkvägen 7', '632 20', 'Eskilstuna', 'Sweden', 'Active'),
    ('Amina Jafari', '20000812-6547', '2000-08-12', 'F', 'amina.jafari@gmail.com', '+46767890123', 'Kungsängsgatan 12', '753 20', 'Uppsala', 'Afghanistan', 'Blocked');

INSERT INTO Users.Alerts
    (CustomerID, AlertType, AlertMessage, Status)
VALUES
    (1, 'Transaction Alert', 'A withdrawal of $500 was made from your account.', 'Read'),
    (2, 'Login Alert', 'Your account was accessed from a new device.', 'Unread'),
    (3, 'Low Balance Alert', 'Your account balance has dropped below $100.', 'Unread'),
    (4, 'Loan Payment Reminder', 'Your loan payment of $1,200 is due in 3 days.', 'Read'),
    (5, 'Credit Card Payment Due', 'Your credit card payment of $300 is due tomorrow.', 'Unread'),
    (6, 'Suspicious Activity', 'Unusual activity detected on your account. Please verify.', 'Read'),
    (7, 'Deposit Confirmation', 'A deposit of $2,000 has been made to your account.', 'Unread'),
    (8, 'Overdraft Warning', 'Your account is overdrawn. Immediate action required.', 'Read'),
    (9, 'Profile Update Alert', 'Your contact information was recently updated.', 'Unread'),
    (10, 'Interest Payment', 'Interest of $50 has been credited to your savings account.', 'Read');

INSERT INTO Bank.Branches
    (BranchName, BranchAddress, BranchPhoneNumber)
VALUES
    ('Stockholm City Bank', 'Drottninggatan 15, 111 51 Stockholm', '+46 8 555 12345'),
    ('Göteborg Finanscenter', 'Avenyn 32, 411 36 Göteborg', '+46 31 789 6789'),
    ('Malmö Bank & Co.', 'Stortorget 7, 211 34 Malmö', '+46 40 222 4567');

INSERT INTO Users.Employees
    (BranchID, Name, Position, Gender, Email, PhoneNumber, Address, PostalCode, City, Country, EmploymentDate, Salary)
VALUES
    (1, 'Erik Johansson', 'Branch Manager', 'M', 'erik.johansson@stockholmbank.se', '+46 70 123 4567', 'Sveavägen 14', '111 57', 'Stockholm', 'Sweden', '2020-06-15', 65000.00),
    (1, 'Fatima Al-Hassan', 'Financial Advisor', 'F', 'fatima.alhassan@stockholmbank.se', '+46 70 234 5678', 'Kungsgatan 34', '111 48', 'Stockholm', 'Sweden', '2021-03-12', 48000.00),
    (1, 'Liam O''Connor', 'Loan Officer', 'M', 'liam.oconnor@stockholmbank.se', '+46 70 345 6789', 'Birger Jarlsgatan 21', '114 34', 'Stockholm', 'Ireland', '2022-07-01', 45000.00),
    (1, 'Chen Wei', 'Customer Service Representative', 'M', 'chen.wei@stockholmbank.se', '+46 70 456 7890', 'Hornsgatan 67', '118 49', 'Stockholm', 'China', '2023-05-23', 37000.00),
    (1, 'Sofia Bergström', 'Accountant', 'F', 'sofia.bergstrom@stockholmbank.se', '+46 70 567 8901', 'Vasagatan 10', '111 20', 'Stockholm', 'Sweden', '2021-11-10', 52000.00),
    (2, 'Anders Lindqvist', 'Branch Manager', 'M', 'anders.lindqvist@gbgfinans.se', '+46 31 100 200', 'Kungsportsavenyn 45', '411 36', 'Göteborg', 'Sweden', '2019-09-20', 64000.00),
    (2, 'Aisha Njeri', 'Investment Analyst', 'F', 'aisha.njeri@gbgfinans.se', '+46 31 111 222', 'Södra Vägen 12', '412 54', 'Göteborg', 'Kenya', '2020-12-01', 51000.00),
    (2, 'Johan Persson', 'Loan Officer', 'M', 'johan.persson@gbgfinans.se', '+46 31 122 333', 'Haga Nygata 9', '411 22', 'Göteborg', 'Sweden', '2023-04-15', 46000.00),
    (2, 'Maria Gonzalez', 'Financial Advisor', 'F', 'maria.gonzalez@gbgfinans.se', '+46 31 133 444', 'Viktoriagatan 6', '411 25', 'Göteborg', 'Spain', '2021-08-29', 49000.00),
    (2, 'Raj Patel', 'Customer Service Representative', 'M', 'raj.patel@gbgfinans.se', '+46 31 144 555', 'Östra Hamngatan 22', '411 10', 'Göteborg', 'India', '2022-10-05', 36000.00),
    (3, 'Karin Svensson', 'Branch Manager', 'F', 'karin.svensson@malmobank.se', '+46 40 200 300', 'Gustav Adolfs torg 8', '211 39', 'Malmö', 'Sweden', '2018-05-14', 63000.00),
    (3, 'Omar El-Sayed', 'Loan Officer', 'M', 'omar.elsayed@malmobank.se', '+46 40 211 400', 'Davidshallsgatan 17', '211 45', 'Malmö', 'Egypt', '2021-02-10', 47000.00),
    (3, 'Emily Carter', 'Financial Advisor', 'F', 'emily.carter@malmobank.se', '+46 40 222 500', 'Föreningsgatan 32', '211 52', 'Malmö', 'USA', '2023-07-18', 49000.00),
    (3, 'Nguyen Hoang', 'Accountant', 'M', 'nguyen.hoang@malmobank.se', '+46 40 233 600', 'Djäknegatan 4', '211 34', 'Malmö', 'Vietnam', '2020-09-22', 53000.00),
    (3, 'Anna Müller', 'Customer Service Representative', 'F', 'anna.muller@malmobank.se', '+46 40 244 700', 'Skomakaregatan 3', '211 36', 'Malmö', 'Germany', '2022-12-01', 35000.00);

UPDATE Bank.Branches SET BranchManagerID ='11' WHERE BranchName = 'Malmö Bank & Co.';
UPDATE Bank.Branches SET BranchManagerID ='6' WHERE BranchName = 'Göteborg Finanscenter';
UPDATE Bank.Branches SET BranchManagerID ='1' WHERE BranchName = 'Stockholm City Bank';
```

We inserted data that is of course fake to showcase how our tables would look like in production with data. (Because Employees table was created after the Branches table, we had to add the Branch Managers for each branch after adding our Employees into the Employees Table as seen in the screenshot above.)

This script is ready from the get-go, just execute it and you'll get a message saying that the script was executed successfully without errors:

```
PRINT 'The script was executed successfully without any errors!'
```

Tables

Schema Users **Primary Key:** 🔑

Tables Customers **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	CustomerID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for customers
2.	-	Name	NVARCHAR(100)	No	-	-	Customer's name
3.	-	SocialSecurityNumber	NVARCHAR(13)	No	UNIQUE	-	Customer's social security number (must be unique)
4.	-	DateOfBirth	DATE	No	-	-	Customer's date of birth
5.	-	Gender	NCHAR(1)	No	(Gender IN ('M', 'F', 'O'))	-	Customer's gender (M = Male, F = Female, O = Others)
6.	-	Email	NVARCHAR(255)	No	UNIQUE	-	Customer's email (must be unique)
7.	-	PhoneNumber	NVARCHAR(20)	No	-	-	Customer's phone number
8.	-	Address	NVARCHAR(255)	No	-	-	Customer's address
9.	-	PostalCode	NVARCHAR(10)	No	-	-	Postal code
10.	-	City	NVARCHAR(50)	No	-	-	City name
11.	-	Country	NVARCHAR(60)	No	-	-	Country name
12.	-	CreatedAt	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the customer was created
13.	-	Status	NVARCHAR(10)	No	CHECK (Status IN ('Active', 'Inactive', 'Blocked'))	-	Customer status (Active, Inactive or Blocked)

Schema Users **Primary Key:** 🔑

Tables Alerts **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	AlertID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for alerts
2.	🔑→	CustomerID	INT	No	FOREIGN KEY	Users.Customers(CustomerID)	References the customer triggering an alert
3.	-	AlertType	NVARCHAR(50)	No	-	-	Declares the type of alert
4.	-	AlertMessage	NVARCHAR(MAX)	No	-	-	Alert messages
5.	-	CreatedAt	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the alert was created
6.	-	Status	NVARCHAR(10)	No	CHECK (Status IN ('Read', 'Unread'))	-	Alerts status (Read or Unread)

Schema Users **Primary Key:** 🔑

Tables Employees **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	EmployeeID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for employees
2.	🔑→	BranchID	INT	No	FOREIGN KEY	Bank.Branches(BranchID)	References the branches the employees works for
3.	-	Name	NVARCHAR(100)	No	-	-	Employee's name
4.	-	Position	NVARCHAR(50)	No	-	-	Employee's position
5.	-	Gender	NVARCHAR(10)	No	CHECK (Gender IN ('M', 'F', 'O'))	-	Customer's gender (M = Male, F = Female, O = Others)
6.	-	Email	NVARCHAR(255)	No	UNIQUE	-	Employee's email (must be unique)

7.	-	PhoneNumber	NVARCHAR(20)	No	-	-	Employee's phone number
8.	-	Address	NVARCHAR(255)	No	-	-	Employee's address
9.	-	PostalCode	NVARCHAR(10)	No	-	-	Postal code
10.	-	City	NVARCHAR(100)	No	-	-	City name
12.	-	Country	NVARCHAR(100)	No	-	-	Country name
13.	-	EmploymentDate	DATE	No	DEFAULT GETDATE()	-	Timestamp when the employee was employed
14.	-	Salary	DECIMAL(18,2)	No	CHECK(Salary >= 0)	-	Employee salary (Ensures that salary is 0 or higher)

Schema Users **Primary Key:** 🔑

Tables Accounts **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	AccountID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for accounts
2.	🔑→	CustomerID	INT	No	FOREIGN KEY	Users.Customers(CustomerID)	References which customers owns the account
3.	🔑→	BranchID	INT	No	FOREIGN KEY	Bank.Branches(BranchID)	References which branch the accounts are created for
4.	🔑→	InterestRateID	INT	No	FOREIGN KEY	Bank.InterestRates(InterestRateID)	References the interest rate applied to the account
5.	-	AccountType	NVARCHAR(30)	No	-	-	Specifies the type of account
6.	-	Balance	DECIMAL(15,2)	No	DEFAULT 0.00	-	Displays the current total balance in the account
7.	-	Currency	NVARCHAR(3)	No	DEFAULT 'SEK'	-	Specifies the currency (Default SEK)

Schema Users **Primary Key:** 🔑

Tables Dispositions **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	DispositionID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for dispositions
2.	🔑→	CustomerID	INT	No	FOREIGN KEY	Users.Customers(CustomerID)	References which customer is associated with the disposition
3.	🔑→	AccountID	INT	No	FOREIGN KEY	Users.Accounts(AccountID)	References which account is associated with the disposition
4.	🔑→	CardID	INT	No	FOREIGN KEY	Bank.Cards(CardID)	References which card is associated with the disposition
5.	-	RelationshipType	NVARCHAR(50)	No	-	-	Specifies the type of relationship between the customer and the account or card
6.	-	DispositionDate	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the disposition was created

Schema Bank **Primary Key:** 🔑

Tables Branches **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	BranchID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for branches
2.	🔑→	BranchManagerID	INT	No	FOREIGN KEY	Users.Employees(EmployeeID)	References this branch's manager
3.	-	BranchName	NVARCHAR(100)	No	-	-	Specifies the name of the branch
4.	-	BranchAddress	NVARCHAR(255)	No	-	-	Branch address
5.	-	BranchPhoneNumber	NVARCHAR(20)	No	-	-	Branch phone number

Schema Bank **Primary Key:** 🔑

Tables InterestRates **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	InterestRateID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for interest rates
2.	-	AccountType	NVARCHAR(100)	No	-	-	Specifies the type of account
3.	-	InterestRate	DECIMAL(5,2)	No	-	-	Specifies the interest rate
4.	-	ValidFromDate	DATE	No	-	-	Specifies the date the interest rate is valid from
5.	-	ValidToDate	DATE	Yes	-	-	Specifies the date the interest rate is valid to

Schema Bank **Primary Key:** 🔑

Tables Cards **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	CardID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for cards
2.	🔑→	AccountID	INT	No	FOREIGN KEY	Users.Accounts(AccountID)	Referenches which account owns this card
3.	-	CardType	NVARCHAR(20)	No	-	-	Specifies the type of card
4.	-	CardNumber	NVARCHAR(16)	No	UNIQUE	-	Customer's own unique card number (must be unique)
5.	-	IssuedDate	DATE	No	-	-	Specifies the date this card was issued
6.	-	ExpiryYear	SMALLINT	No	-	-	Specifies the date the card expires
7.	-	Status	NVARCHAR(10)	No	CHECK (Status IN ('Active', 'Blocked', 'Expired'))	-	Cards status (Active, Blocked or Expired)

Schema Bank **Primary Key:** 🔑

Tables LoanPayments **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	PaymentID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for loan payments
2.	🔑→	LoanID	INT	No	FOREIGN KEY	Bank.Loans(LoanID)	References which loan the loan payment is for
3.	-	PaymentDate	DATETIME	No	DEFAULT GETDATE()	-	Specifies when the payment was made
4.	-	PaymentAmount	DECIMAL(10,2)	No	-	-	Specifies the total payment amount

5.	-	RemainingBalance	DECIMAL(10,2)	No	-	-	Specifies the remaining balance to be paid
6.	-	PaymentMethod	NVARCHAR(20)	No	CHECK (PaymentMethod IN ('Bank Transfer', 'Cash', 'Credit Card', 'Check', 'Other'))	-	Payment method (Bank Transfer, Cash, Credit Card, Check or Other)

Schema Bank **Primary Key:** 🔑

Tables Loans **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	LoanID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for loans
2.	🔑→	CustomerID	INT	No	FOREIGN KEY	Users.Customers(CustomerID)	References which customer has a loan
3.	🔑→	InterestRateID	INT	No	FOREIGN KEY	Bank.InterestRates(InterestRateID)	References which interest rate of the loan
4.	-	LoanType	NVARCHAR(20)	No	-	-	Specifies the loan type
5.	-	LoanAmount	DECIMAL(18,2)	No	-	-	Specifies the total loan amount
6.	-	LoanTerm	NVARCHAR(10)	No	-	-	Specifies the loans length of time to be repaid
7.	-	InterestRate	DECIMAL(5,2)	No	-	-	Specifies the interest rate for the loan
8.		LoanStartDate	DATE	No	-	-	Specifies when the customer took the loan
9.		LoanEndDate	DATE	No	-	-	Specifies when the loan is paid off
10.		MonthlyPayment	DECIMAL(18,2)	No	-	-	Specifies the monthly payment for the loan
11.		Status	NVARCHAR(10)	No	CHECK (Status IN ('Active', 'Closed', 'Defaulted'))	-	Loan status (Active, Closed or Defaulted)

Schema Security **Primary Key:** 🔑

Tables AuditLogs **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	AuditLogID	INT	No	PRIMARY KEY IDENTITY(1,1)	.	Unique identifiers for audit logs
2.	🔑→	EmployeeID	INT	No	FOREIGN KEY	Users.Employees(EmployeeID)	References which employee made an action in the database
3.	-	Action	NVARCHAR(255)	No	-	-	Specifies what type of action was made
4.	-	Timestamp	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the audit was logged
5.	-	AffectedTable	NVARCHAR(255)	No	-	-	Specifies which table was changed
6.	-	AffectedRowID	INT	Yes	-	-	Specifies the rows id that was changed
7.	-	OldValue	NVARCHAR(MAX)	Yes	-	-	Specifies the old value of the column/table
8.	-	NewValue	NVARCHAR(MAX)	Yes	-	-	Specifies the new value of the column/table

Schema Finance **Primary Key:** 🔑

Tables Transactions **Foreign Key:** 🔑→

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	TransactionID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for transactions
2.	🔑→	AccountID	INT	No	FOREIGN KEY	Users.Accounts(AccountID)	References which account made the transaction
3.	-	TransactionType	NVARCHAR(20)	No	CHECK (TransactionType IN ('Deposit',	-	Specifies the transaction type

					'Withdrawal', 'Transfer'))		
4.	-	Amount	DECIMAL(18,2)	No	-	-	Specifies the amount of the transaction
5.	-	TransactionDate	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the transaction was made
6.	-	TransactionMethod	NVARCHAR(50)	No	CHECK (TransactionMethod IN ('Cash', 'Card', 'Online Banking', 'Wire Transfer', 'Mobile Payment'))	-	Transaction method (Cash, Card, Online Banking, Wire Transfer or Mobile Payment)
7.	-	Description	NVARCHAR(MAX)	Yes	-	-	Description of the transaction

Schema Finance

Tables TransactionLogs

Primary Key: 🔑

Foreign Key: 🔑➞

	Key	Name	Data Type	Null	Attributes	References	Description
1.	🔑	LogID	INT	No	PRIMARY KEY IDENTITY(1,1)	-	Unique identifier for transaction logs
2.	🔑➞	TransactionID	INT	No	FOREIGN KEY	Finance.Transactions(TransactionID)	References the transaction that was logged
3.	🔑➞	EmployeeID	INT	No	FOREIGN KEY	Users.Employees(EmployeeID)	References the employee that administered this transaction
4.	-	LogMessage	NVARCHAR(MAX)	No	-	-	Description of the log
5.	-	LogTimestamp	DATETIME	No	DEFAULT GETDATE()	-	Timestamp when the transaction was logged

Table Overview

We will now display how each table looks with the fake data generated by ChatGPT.

Users.Customers

	CustomerID	Name	SocialSecurityNumber	DateOfBirth	Gender	Email	PhoneNumber	Address	PostalCode	City	Country	CreatedAt	Status
1	1	Johan Svensson	19880415-7845	1988-04-15	M	johan.svensson@gmail.com	+46738911234	Storgatan 12	114 56	Stockholm	Sweden	2025-04-30 18:38:45.647	Active
2	2	Fatima Hussein	19950610-4521	1995-06-10	F	fatima.hussein@yahoo.com	+46763224567	Bergsgatan 23	211 12	Malmö	Iraq	2025-04-30 18:38:45.647	Active
3	3	Anders Karlsson	20010821-9856	2001-08-21	M	anders.karlsson@hotmail.com	+46782347654	Linnégatan 45	413 04	Göteborg	Sweden	2025-04-30 18:38:45.647	Inactive
4	4	Sofia Zhang	20050411-5673	2005-04-11	F	sofia.zhang@gmail.com	+46765438912	Kungsgatan 8	753 20	Uppsala	China	2025-04-30 18:38:45.647	Active
5	5	Oskar Berg	19991202-3456	1999-12-02	M	oskar.berg@gmail.com	+46734598761	Drottninggatan 30	803 10	Gävle	Sweden	2025-04-30 18:38:45.647	Blocked
6	6	Lena Persson	19830527-1234	1983-05-27	F	lena.persson@gmail.com	+46768765432	Huvudvägen 5	541 45	Skövde	Sweden	2025-04-30 18:38:45.647	Active
7	7	Ahmed Ali	20020814-9872	2002-08-14	M	ahmed.ali@gmail.com	+46769876543	Rosengårdsvägen 6	213 66	Malmö	Somalia	2025-04-30 18:38:45.647	Inactive
8	8	Emma Lindström	19970303-7654	1997-03-03	F	emma.lindstrom@gmail.com	+46765412389	Torsgatan 18	411 03	Göteborg	Sweden	2025-04-30 18:38:45.647	Blocked
9	9	David Brown	19900129-4321	1990-01-29	M	david.brown@gmail.com	+46768901234	Sveavägen 60	113 59	Stockholm	USA	2025-04-30 18:38:45.647	Active
10	10	Aliyah Mohamed	20070930-6543	2007-09-30	F	aliyah.mohamed@gmail.com	+46769871234	Lantmannagatan 9	214 50	Malmö	Eritrea	2025-04-30 18:38:45.647	Inactive
11	11	Elliot Larsson	20100205-8765	2010-02-05	O	elliot.larsson@gmail.com	+46769876521	Vasagatan 3	722 11	Västerås	Sweden	2025-04-30 18:38:45.647	Active
12	12	Isabella Rossi	19920418-4323	1992-04-18	F	isabella.rossi@gmail.com	+46768904321	Östra Hamngatan 4	411 10	Göteborg	Italy	2025-04-30 18:38:45.647	Blocked
13	13	Kristian Nilsson	20060317-5432	2006-03-17	M	kristian.nilsson@gmail.com	+46765498721	Brunnsgatan 19	111 38	Stockholm	Sweden	2025-04-30 18:38:45.647	Inactive
14	14	Morgan Sjöberg	19891224-3214	1989-12-24	O	morgan.sjoberg@gmail.com	+46769871239	Björkvägen 7	632 20	Eskilstuna	Sweden	2025-04-30 18:38:45.647	Active
15	15	Amina Jafari	20080812-6547	2008-08-12	F	amina.jafari@gmail.com	+46767890123	Kungsängsgatan 12	753 20	Uppsala	Afghanistan	2025-04-30 18:38:45.647	Blocked

Users.Alerts

	AlertID	CustomerID	AlertType	AlertMessage	CreatedAt	Status
1	1	1	Transaction Alert	A withdrawal of $500 was made from your account.	2025-04-30 18:38:45.653	Read
2	2	2	Login Alert	Your account was accessed from a new device.	2025-04-30 18:38:45.653	Unread
3	3	3	Low Balance Alert	Your account balance has dropped below $100.	2025-04-30 18:38:45.653	Unread
4	4	4	Loan Payment Reminder	Your loan payment of $1,200 is due in 3 days.	2025-04-30 18:38:45.653	Read
5	5	5	Credit Card Payment Due	Your credit card payment of $300 is due tomorrow.	2025-04-30 18:38:45.653	Unread
6	6	6	Suspicious Activity	Unusual activity detected on your account. Please...	2025-04-30 18:38:45.653	Read
7	7	7	Deposit Confirmation	A deposit of $2,000 has been made to your account.	2025-04-30 18:38:45.653	Unread
8	8	8	Overdraft Warning	Your account is overdrawn. Immediate action requi...	2025-04-30 18:38:45.653	Read
9	9	9	Profile Update Alert	Your contact information was recently updated.	2025-04-30 18:38:45.653	Unread
10	10	10	Interest Payment	Interest of $50 has been credited to your savings a...	2025-04-30 18:38:45.653	Read

Users.Employees

	EmployeeID	BranchID	Name	Position	Gender	Email	PhoneNumber	Address	PostalCode	City	Country	EmploymentDate	Salary
1	1	1	Erik Johansson	Branch Manager	M	erik.johansson@stockholmbank.se	+46 70 123 4567	Sveavägen 14	111 57	Stockholm	Sweden	2020-06-15	65000.00
2	2	1	Fatima Al-Hassan	Financial Advisor	F	fatima.alhassan@stockholmbank.se	+46 70 234 5678	Kungsgatan 34	111 48	Stockholm	Sweden	2021-03-12	48000.00
3	3	1	Liam O'Connor	Loan Officer	M	liam.oconnor@stockholmbank.se	+46 70 345 6789	Birger Jarlsgatan 21	114 34	Stockholm	Ireland	2022-07-01	45000.00
4	4	1	Chen Wei	Customer Service Representative	M	chen.wei@stockholmbank.se	+46 70 456 7890	Hornsgatan 67	118 49	Stockholm	China	2023-05-23	37000.00
5	5	1	Sofia Bergström	Accountant	F	sofia.bergstrom@stockholmbank.se	+46 70 567 8901	Vasagatan 10	111 20	Stockholm	Sweden	2021-11-10	52000.00
6	6	2	Anders Lindqvist	Branch Manager	M	anders.lindqvist@gbgfinans.se	+46 31 100 200	Kungsportsavenyn 45	411 36	Göteborg	Sweden	2019-09-20	64000.00
7	7	2	Aisha Njeri	Investment Analyst	F	aisha.njeri@gbgfinans.se	+46 31 111 222	Södra Vägen 12	412 54	Göteborg	Kenya	2020-12-01	51000.00
8	8	2	Johan Persson	Loan Officer	M	johan.persson@gbgfinans.se	+46 31 122 333	Haga Nygata 9	411 22	Göteborg	Sweden	2023-04-15	46000.00
9	9	2	Maria Gonzalez	Financial Advisor	F	maria.gonzalez@gbgfinans.se	+46 31 133 444	Viktoriagatan 6	411 25	Göteborg	Spain	2021-08-29	49000.00
10	10	2	Raj Patel	Customer Service Representative	M	raj.patel@gbgfinans.se	+46 31 144 555	Östra Hamngatan 22	411 10	Göteborg	India	2022-10-05	36000.00
11	11	3	Karin Svensson	Branch Manager	F	karin.svensson@malmobank.se	+46 40 200 300	Gustav Adolfs torg 8	211 39	Malmö	Sweden	2018-05-14	63000.00
12	12	3	Omar El-Sayed	Loan Officer	M	omar.elsayed@malmobank.se	+46 40 211 400	Davidshallsgatan 17	211 45	Malmö	Egypt	2021-02-10	47000.00
13	13	3	Emily Carter	Financial Advisor	F	emily.carter@malmobank.se	+46 40 222 500	Föreningsgatan 32	211 52	Malmö	USA	2023-07-18	49000.00
14	14	3	Nguyen Hoang	Accountant	M	nguyen.hoang@malmobank.se	+46 40 233 600	Djäknegatan 4	211 34	Malmö	Vietnam	2020-09-22	53000.00
15	15	3	Anna Müller	Customer Service Representative	F	anna.muller@malmobank.se	+46 40 244 700	Skomakaregatan 3	211 36	Malmö	Germany	2022-12-01	35000.00

Users.Accounts

	AccountID	CustomerID	BranchID	InterestRateID	AccountType	Balance	Currency
1	1	1	2	1	Savings	12500.00	SEK
2	2	2	1	2	Current	4570.50	SEK
3	3	3	3	3	Fixed Deposit	30000.00	SEK
4	4	4	1	2	Savings	9800.00	SEK
5	5	5	2	1	Current	2140.75	SEK
6	6	6	3	3	Savings	15250.00	SEK
7	7	7	1	2	Fixed Deposit	50000.00	SEK
8	8	8	2	1	Current	6350.00	SEK
9	9	9	3	2	Savings	8730.00	SEK
10	10	10	1	3	Fixed Deposit	45000.00	SEK
11	11	11	2	2	Savings	2760.00	SEK
12	12	12	3	1	Current	3900.00	SEK
13	13	13	1	3	Savings	7150.00	SEK
14	14	14	2	2	Current	4870.00	SEK
15	15	15	3	1	Fixed Deposit	100000.00	SEK

Users.Dispositions

	DipositionID	CustomerID	AccountID	CardID	RelationshipType	DipositionDate
1	1	1	1	1	Owner	2023-01-15 00:00:00.000
2	2	2	2	2	Authorized User	2023-03-10 00:00:00.000
3	3	3	3	3	Joint Holder	2022-11-08 00:00:00.000
4	4	4	4	4	Owner	2024-02-25 00:00:00.000
5	5	5	5	5	Power of Attorney	2023-05-12 00:00:00.000
6	6	6	6	6	Authorized User	2023-07-03 00:00:00.000
7	7	7	7	7	Owner	2023-09-14 00:00:00.000
8	8	8	8	8	Joint Holder	2022-12-29 00:00:00.000
9	9	9	9	9	Owner	2023-10-20 00:00:00.000
10	10	10	10	10	Authorized User	2024-01-05 00:00:00.000
11	11	11	11	11	Owner	2023-06-18 00:00:00.000
12	12	12	12	12	Power of Attorney	2023-04-09 00:00:00.000
13	13	13	13	13	Authorized User	2023-08-23 00:00:00.000
14	14	14	14	14	Owner	2022-10-31 00:00:00.000
15	15	15	15	15	Joint Holder	2023-11-11 00:00:00.000

Bank.Branches

	BranchID	BranchManagerID	BranchName	BranchAddress	BranchPhoneNumber
1	1	1	Stockholm City Bank	Drottninggatan 15, 111 51 Stockholm	+46 8 555 12345
2	2	6	Göteborg Finanscenter	Avenyn 32, 411 36 Göteborg	+46 31 789 6789
3	3	11	Malmö Bank & Co.	Stortorget 7, 211 34 Malmö	+46 40 222 4567

Bank.InterestRate

	InterestRateID	AccountType	InterestRate	ValidFromDate	ValidToDate
1	1	Savings Account	1.50	2024-01-01	2025-12-31
2	2	Fixed Deposit - 6 months	2.25	2024-01-01	2024-06-30
3	3	Fixed Deposit - 1 year	2.75	2024-01-01	2024-12-31
4	4	Fixed Deposit - 3 years	3.25	2024-01-01	2026-12-31
5	5	Fixed Deposit - 5 years	3.75	2024-01-01	2028-12-31
6	6	Checking Account	0.10	2024-01-01	2026-12-31
7	7	Business Account	0.75	2024-01-01	2025-12-31
8	8	Premium Savings Account	2.00	2024-01-01	2026-06-30
9	9	Youth Savings Account	1.75	2024-01-01	2025-06-30
10	10	Retirement Savings Account	2.50	2024-01-01	2029-12-31
11	11	Mortgage Loan	4.50	2024-01-01	2026-12-31
12	12	Personal Loan	6.25	2024-01-01	2025-06-30
13	13	Car Loan	5.00	2024-01-01	2025-12-31
14	14	Credit Card	15.99	2024-01-01	2026-12-31

Bank.Cards

	CardID	AccountID	CardType	CardNumber	IssuedDate	ExpiryYear	Status
1	1	1	Debit	4539123498761234	2021-03-10	2026	Active
2	2	1	Credit	5243678912345678	2020-07-15	2025	Active
3	3	2	Debit	4765823471982345	2021-06-11	2026	Blocked
4	4	2	Credit	5398123456239876	2019-01-15	2024	Expired
5	5	3	Debit	4029384712983471	2022-08-05	2027	Active

Bank.Loans

	LoanID	CustomerID	InterestRateID	LoanType	LoanAmount	LoanTerm	InterestRate	LoanStartDate	LoanEndDate	MonthlyPayment	Status
1	1	3	2	Home	350000.00	240M	3.25	2020-05-15	2040-05-15	1916.42	Active
2	2	5	1	Car	25000.00	60M	4.10	2021-03-10	2026-03-10	460.59	Closed
3	3	7	4	Personal	10000.00	36M	5.20	2022-07-01	2025-07-01	300.21	Active
4	4	2	3	Business	150000.00	120M	4.75	2020-01-20	2030-01-20	1570.36	Defaulted
5	5	1	5	Education	30000.00	84M	3.75	2019-09-01	2026-09-01	408.33	Closed
6	6	9	2	Car	18000.00	48M	4.25	2021-11-12	2025-11-12	407.12	Active
7	7	4	1	Home	420000.00	360M	3.00	2018-06-01	2048-06-01	1772.07	Active
8	8	6	3	Business	200000.00	180M	4.95	2022-01-15	2037-01-15	1580.89	Active
9	9	10	5	Personal	15000.00	36M	5.80	2023-03-01	2026-03-01	454.63	Closed
10	10	8	4	Education	25000.00	60M	4.50	2020-09-20	2025-09-20	466.65	Defaulted
11	11	2	2	Car	22000.00	72M	3.95	2019-12-01	2025-12-01	341.87	Active
12	12	3	3	Business	120000.00	84M	5.10	2021-08-10	2028-08-10	1691.22	Active
13	13	1	5	Home	275000.00	180M	3.75	2018-02-01	2033-02-01	2002.34	Closed
14	14	6	1	Education	18000.00	48M	4.20	2022-05-10	2026-05-10	408.97	Active
15	15	4	2	Personal	12000.00	24M	5.60	2021-01-01	2023-01-01	530.75	Closed

Bank.LoanPayments

	PaymentID	LoanID	PaymentDate	PaymentAmount	RemainingBalance	PaymentMethod
1	1	1	2023-01-01 00:00:00.000	684.67	317694.83	Other
2	2	1	2023-02-01 00:00:00.000	1474.72	316220.11	Check
3	3	2	2023-01-01 00:00:00.000	2486.49	44384.04	Credit Card
4	4	2	2023-01-30 00:00:00.000	1992.55	42391.50	Credit Card
5	5	2	2023-02-27 00:00:00.000	1558.33	40833.17	Check

Security.AuditLogs

	AuditLogID	EmployeeID	Action	Timestamp	AffectedTable	AffectedRowID	OldValue	NewValue
1	1	6	UPDATE	2025-04-05 10:23:45.000	Users.Customers	12	Email: old@example.com	Email: new@example.com
2	2	2	INSERT	2025-04-04 14:12:22.000	Finance.Accounts	45	NULL	New record created
3	3	14	DELETE	2025-04-06 09:00:00.000	Finance.Loans	7	LoanID: 12, Amount: 10000	NULL
4	4	1	UPDATE	2025-04-03 12:44:55.000	Users.Customers	33	Phone: +46 70 111 1111	Phone: +46 70 222 2222
5	5	12	INSERT	2025-04-06 08:12:10.000	Bank.Transactions	88	NULL	Initial transaction logged

Finance.Transacitons

	TransactionID	AccountID	TransactionType	Amount	TransactionDate	TransactionMethod	Description
1	1	1	Deposit	3299.45	2023-01-12 00:00:00.000	Wire Transfer	Salary Payment
2	2	2	Withdrawal	550.20	2023-02-10 00:00:00.000	Cash	ATM Withdrawal
3	3	3	Transfer	1023.99	2023-03-05 00:00:00.000	Mobile Payment	Utility Bill
4	4	4	Deposit	4420.00	2023-04-08 00:00:00.000	Online Banking	Freelance Payment
5	5	5	Withdrawal	920.10	2023-05-15 00:00:00.000	Card	Grocery Shopping

Finance.TransactionLogs

	LogID	TransactionID	EmployeeID	LogMessage	LogTimestamp
1	1	1	2	Verified deposit and updated balance.	2023-01-12 09:34:12.000
2	2	2	1	Processed ATM withdrawal request.	2023-02-10 11:20:45.000
3	3	3	3	Checked transfer routing details.	2023-03-05 15:50:30.000
4	4	4	2	Confirmed incoming salary deposit.	2023-04-08 10:05:22.000
5	5	5	1	Withdrawal flagged for large amount check.	2023-05-15 13:14:10.000

Third Normal Form (3NF)

We designed this database to follow the rules of Third Normal Form (3NF), which means the structure avoids unnecessary repetition and keeps the data organized in a smart way.

- All fields in the tables store only **one piece of information** (1NF).
- Every field that isn't a primary key depends **completely** on the primary key of the table (2NF).
- There are **no fields that depend on other non-key fields** (3NF), meaning each piece of information is stored in the right place and not repeated across different tables.

Long-Term Improvements

We were tasked with a relatively simple task, to create a database for a bank. While we think that we've managed to create a database that fits a bank, there are of course still things that could be improved but here are some of them:

Security Enhancements

- Encrypting sensitive data such as Cardnumber, SocialSecurityNumber by using hashing or encryption functions
- Adding UserID, IP Address for more detailed tracking in the Audit log

Additional Tables

These are the tables we think would work for the customers, in a realistic setting where the customer may want to have more or fewer tables, the customer would be involved during the process and add or remove tables depending on their needs. With closer contact with the customer, creating this model and database to it's completed state would be more time efficient.

Stored Procedures, Indexes and Functions

The next step is creating a fully functional script with different procedures and functions that is needed for a bank database to work. Creating indexes to make reoccurring columns/texts load faster.

Files

These are the files used for this project:

- Bank_ER_Diagram_Conceptual – The conceptual model of this database
- Bank_ER_Diagram – The more detailed logical model of this database
- Bank_Script – The main code, just execute this file into your SSMS and you're good to go.
- Bank_Dokumentation – Documentation for this project (This document)

This concludes the documentation. We appreciate your time and hope this project meets your expectations. If you have any questions or require further assistance, please feel free to reach out.