

# Project OPbank - Best practice

Last updated: 2025-05-01

Author: *Ottilia Pettersson*

Version: 1

## 1. Description of the system

The client OPbank is a financial institution that provides banking services to customers. We have started to help the client implement a customer and account management system. The system so far manages information about customers, accounts, cards, transactions, loans, and login attempts.

The database is designed to comply with 3NF through normalisation, which means that transitive dependencies have been eliminated by breaking out columns into separate tables and then linking them to the relevant table. This normalisation ensures that the database is efficient, reduces redundancy and provides a clear structure. 3NF will continue to be implemented in the database as it is further developed.

In the database we have a many to many relationship between customer and account because customers in the bank can have common accounts, this many to many relationship is managed by the linkage table disposition.

## 2. Structure of the database (Tables and columns)

### *Country*

<i>Column</i>	<i>Type</i>	<i>Description</i>
CountryID (PK)	INT	Unique identifier for the country
CountryName	NVARCHAR(30)	Name of the country

### *City*

<i>Column</i>	<i>Type</i>	<i>Description</i>
CityID (PK)	INT	Unique identifier for the city
CountryID (FK)	INT	Reference to the country
CityName	NVARCHAR(30)	Name of the city

### *PostalCode*

<i>Column</i>	<i>Type</i>	<i>Description</i>
PostalCodeID (PK)	INT	Unique identifier for the postal code
CityID (FK)	INT	Reference to the city
PostalCode	NVARCHAR(20)	Postal code

### *Address*

<i>Column</i>	<i>Type</i>	<i>Description</i>
---------------	-------------	--------------------

AddressID (PK)	INT	Unique identifier for the address
PostalCodeID (FK)	INT	Reference to the postal code
Address	NVARCHAR(50)	Street address

### ***Customer***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CustomerID (PK)	INT	Unique identifier for the customer
AddressID (FK)	INT	Reference to the customers address
FirstName	NVARCHAR(25)	Customers first name
LastName	NVARCHAR(25)	Customers last name
BirthDate	DATE	Customers date of birth
SocialSecurityNumberSalt	NVARCHAR(100)	Salt for social security number
SocialSecurityNumberHash	NVARCHAR(100)	Hashed social security number
EmailAddress	NVARCHAR(50)	Customers email address
PhoneNumber	NVARCHAR(25)	Customers phone number
CreatedAt	DATETIME	Date of customer registration

### ***LoginAttempts***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
AttemptID (PK)	INT	Unique identifier for the login attempt
CustomerID (FK)	INT	Reference to the customer
AuthMethodID (FK)	INT	Reference to the authentication method
LoginStatusID (FK)	INT	Reference to the login status
IPAddress	NVARCHAR(50)	IP address of the login attempt
RequestTime	DATETIME	Time of the login request
AttemptTime	DATETIME	Time of the login attempt
BankIDVerified	BIT	Indicates if BankID verification was used

### ***AuthMethod***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
AuthMethodID (PK)	INT	Unique identifier for the authentication method
AuthMethod	NVARCHAR(30)	Authentication method

### **LoginStatus**

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoginStatusID (PK)	INT	Unique identifier for the login status

StatusCode	NVARCHAR(10)	Code for the login status
StatusMessage	NVARCHAR(50)	Message describing the login status

### ***CreditScore***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CreditScoreID (PK)	INT	Unique identifier for the credit score
CustomerID (FK)	INT	Reference to the customer
CreditLevelID (FK)	INT	Reference to the credit level
CreditChecked	DATE	Date the credit was checked

### ***CreditLevel***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CreditLevelID (PK)	INT	Unique identifier for the credit level
CreditLevel	NVARCHAR(15)	Credit level
Score	INT	Credit score value

### ***Loan***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoanID (PK)	INT	Unique identifier for the loan
CustomerID (FK)	INT	Reference to the customer
AccountID (FK)	INT	Reference to the account
LoanStatusID (FK)	INT	Reference to the loan status
LoanTypeID (FK)	INT	Reference to the loan type
LoanAmount	DECIMAL(18,2)	Loan amount
InterestRate	DECIMAL(5,2)	Interest rate of the loan
LoanPeriod	INT	Loan duration in months
StartDate	DATE	Loan start date
EndDate	DATE	Loan end date

### ***LoanType***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoanTypeID (PK)	INT	Unique identifier for the loan type
LoanType	NVARCHAR(20)	Type of loan

### ***LoanStatus***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
----------------------	--------------------	---------------------------

LoanStatusID (PK)	INT	Unique identifier for the loan status
LoanStatus	NVARCHAR(20)	Status of the loan

### ***LoanPayment***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoanPaymentID (PK)	INT	Unique identifier for the loan payment
LoanID (FK)	INT	Reference to the loan
LoanPaymentStatusID (FK)	INT	Reference to the payment status
LoanPaymentMethodID (FK)	INT	Reference to the payment method
InterestAmount	DECIMAL(10,2)	Interest amount of the payment
PrincipalAmount	DECIMAL(10,2)	Principal amount of the payment
FeeAmount	DECIMAL(10,2)	Fee amount of the payment
TotalAmount	DECIMAL(10,2)	Total amount of the payment
DueDate	DATE	Due date for the payment
PaymentDate	DATE	Date the payment was made

### ***LoanPaymentStatus***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoanPaymentStatusID (PK)	INT	Unique identifier for the payment status
LoanPaymentStatus	NVARCHAR(20)	Status of the payment

### ***LoanPaymentMethod***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
LoanPaymentMethodID (PK)	INT	Unique identifier for the payment method
LoanPaymentMethod	NVARCHAR(20)	Method of payment

### ***Disposition***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
DispositionID (PK)	INT	Unique identifier for the disposition
CustomerID (FK)	INT	Reference to the customer
AccountID (FK)	INT	Reference to the account
CardID (FK)	INT	Reference to the card

### ***Card***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CardID (PK)	INT	Unique identifier for the card
AccountID (FK)	INT	Reference to the account
CardStatusID (FK)	INT	Reference to the card status
CardTypeID (FK)	INT	Reference to the card type
CardNumber	NVARCHAR(25)	Unique card number
CreatedDate	DATE	Date the card was created
ExpiryDate	DATE	Card expiration date
CVV2Salt	NVARCHAR(100)	Salt for CVV2
CVV2Hash	NVARCHAR(100)	Hashed CVV2
PincodeSalt	NVARCHAR(100)	Salt for PIN code
PincodeHash	NVARCHAR(100)	Hashed PIN code

### ***CardStatus***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CardStatusID (PK)	INT	Unique identifier for the card status
CardStatus	NVARCHAR(20)	Status of the card

### ***CardType***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
CardTypeID (PK)	INT	Unique identifier for the card type
CardType	NVARCHAR(20)	Type of card

### ***Account***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
AccountID (PK)	INT	Unique identifier for the account
CustomerID (FK)	INT	Reference to the customer
AccountStatusID (FK)	INT	Reference to the account status
AccountTypeID (FK)	INT	Reference to the account type
AccountNumber	NVARCHAR(50)	Unique account number
Balance	DECIMAL(18,6)	Current balance of the account
CreatedAt	DATETIME	Date of account creation

### ***AccountStatus***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
----------------------	--------------------	---------------------------

AccountStatusID (PK)	INT	Unique identifier for the account status
AccountStatus	NVARCHAR(20)	Status of the account

#### ***AccountType***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
AccountTypeID (PK)	INT	Unique identifier for the account type
AccountType	NVARCHAR(20)	Type of account

#### ***Transaction***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
TransactionID (PK)	INT	Unique identifier for the transaction
AccountID (FK)	INT	Reference to the account
TransactionStatusID (FK)	INT	Reference to the transaction status
TransactionTypeID (FK)	INT	Reference to the transaction type
ReceiverAccount	NVARCHAR(50)	Receivers account number
Amount	DECIMAL(18,2)	Transaction amount
Date	DATETIME	Date of the transaction
Description	NVARCHAR(50)	Description of the transaction

#### ***TransactionStatus***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
TransactionStatusID (PK)	INT	Unique identifier for the transaction status
TransactionStatus	NVARCHAR(20)	Status of the transaction

#### ***TransactionType***

<b><i>Column</i></b>	<b><i>Type</i></b>	<b><i>Description</i></b>
TransactionTypeID (PK)	INT	Unique identifier for the transaction type
TransactionType	NVARCHAR(20)	Type of transaction

### **3. Functionality and Implementation**

#### ***STORED PROCEDURE “CreateCustomer”***

This stored procedure creates new customers and adds them to OPbank's database in a secure and structured way. The procedure receives the customer's personal data and manages the secure storage of the sensitive information through encryption. The social security number is hashed and salted with SHA-256 before being stored in the database. This means that the social security number is encrypted and the social security number is not stored in the database. If something goes wrong it is caught and an error message shows. Once the user has been created and added to the database, we get the message: "Customer created successfully". This procedure

ensures that customer data is handled securely and provides a robust method for adding new customers to the system.

#### ***STORED PROCEDURE “CreateCard”***

This stored procedure is similar to ‘create customer’ but this creates customers cards in a secure way and adds them to the database. The procedure handles the registration of card details and encrypts sensitive information as CVV2 code and pin code to protect customer data. Just like in the previous stored procedure will sensitive information such as CVV2 and pin code be hashed and salted with SHA-256 to protect customer data. This makes the data encrypted before being stored in the database. If something goes wrong it is caught and an error message shows. Once the user has been created and added to the database, we get the message: "Customer created successfully". This procedure ensures that card data is handled securely and provides a reliable method for registering new cards in the database.

#### **4. Optimization and Future Improvements**

##### ***Optimization***

- The database is normalized to at least 3NF with separate tables for addresses, status and types.
- The code includes indexes for better performance and it will continue to be added to the code as we develop.
- We use unique constraints to avoid duplicate data.
- We are constantly working on security to be a safe bank and protect our customers' data stored in our database. When developing the database, we keep security in mind and always implement encryption solutions for data that should not be visible.

##### ***Future Improvements***

- What has been created now in the database is just the basis, OPbank will constantly develop more and has a lot left to create until it is a complete database for a bank. The next step is to expand the database so we can manage employees.
- The encryption currently in the database is only temporary and will be replaced with a stronger variant to minimize the risk of sensitive data being disseminated in the scenario of a breach.
- Continue implementing indexes in code to prevent slow performance.
- We are working to reach even higher normal form than 3NF.