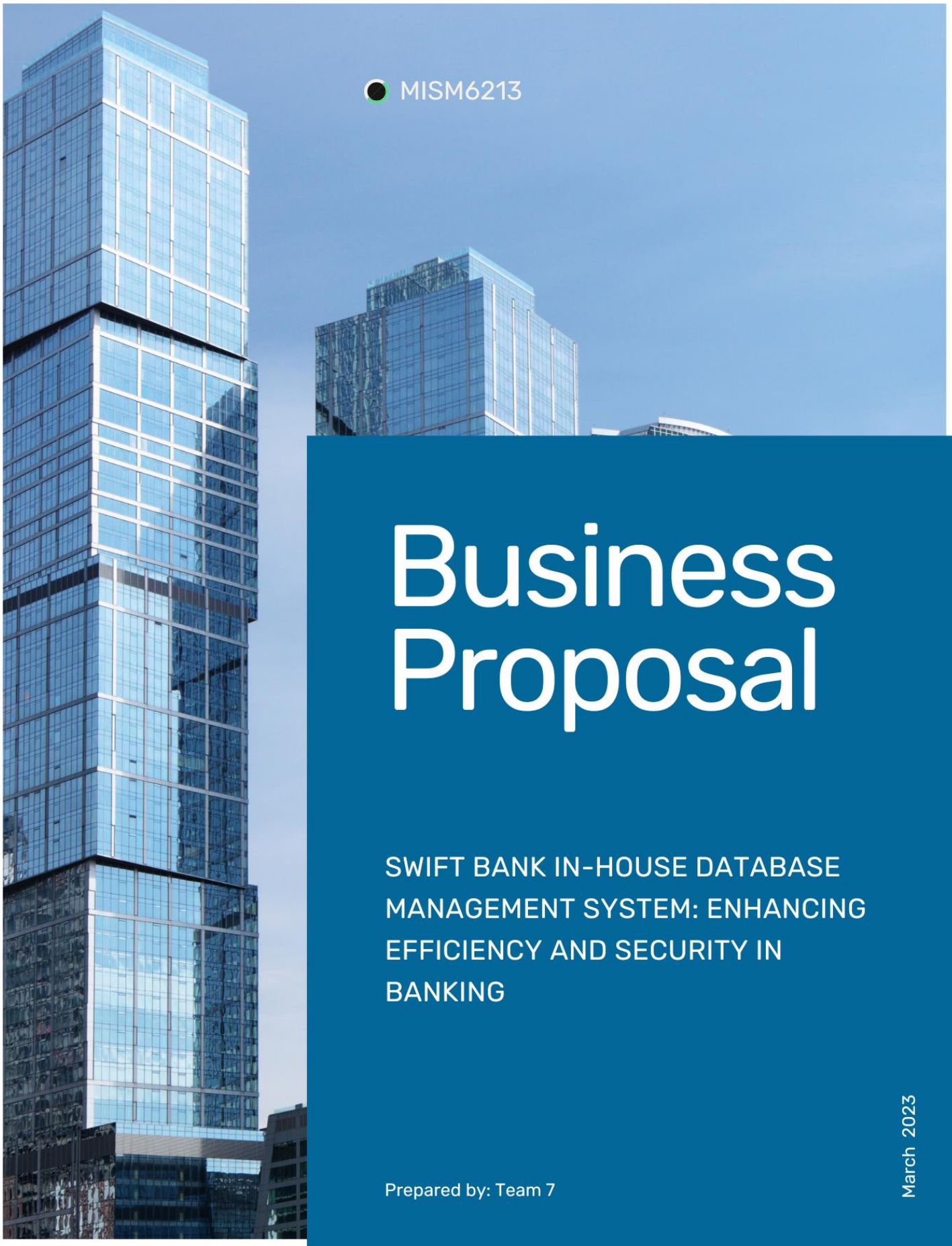


MISM6213

Business Proposal

SWIFT BANK IN-HOUSE DATABASE MANAGEMENT SYSTEM: ENHANCING EFFICIENCY AND SECURITY IN BANKING

Prepared by: Team 7

March 2023

Table of Contents

Phase 1: Project Proposal

Problem Statement

This project proposal centers around Swift Bank, a nascent private company seeking to develop an in-house bank database management system that can securely store and manage sensitive financial information, process real-time transactions, and provide valuable insights through data analytics and reporting.

Efficient and dependable bank management systems are increasingly essential in today's financial landscape as financial institutions recognize the importance of a robust database system to oversee their customers' financial information and transactions. According to McKinsey & Co, successful legacy modernization involves leaving data at the edge and building a flexible data platform over time, reducing IT costs and complexity while boosting productivity.[1] Without a well-designed database system, institutions risk inaccuracies, poor customer service, and reputational damage, leading to a loss of business. By developing the system in-house, Swift Bank will have complete control over its design and functionality, ensuring that it meets the organization's specific requirements.

To conclude, a database management system is essential to the domain of banking and financial services because it allows financial institutions to store, manage, and process large volumes of sensitive financial information accurately and efficiently. Building a well-designed bank database management system is crucial for Swift Bank to thrive and stay competitive in the dynamic banking industry. As a new private bank, it will need to establish a reputation for providing efficient, reliable, and secure financial services to its customers. A well-designed bank management system can help to achieve these goals by providing the necessary tools and capabilities to manage customer account, balances, and transactions.

Functionality

The Swift Bank Management System is designed to serve a wide range of stakeholders including bank employees such as tellers, managers, and customer service representatives who need to access and manage customer account information. As well, customers themselves will be able to use the system for self-service tasks such as checking their account balances, reviewing transaction history, and performing transactions like deposits, withdrawals, and transfers. Additionally, another relevant stakeholder would be regulatory authorities who may also access the system to verify compliance with regulatory requirements.

Overall, the Swift Bank Management System database provides a comprehensive management system for end-users to handle customer and account management, transaction processing, and merchant data in support of the bank's day-to-day operations.

[1] McKinsey & Company. (2019). How banks can achieve next-generation legacy modernization. McKinsey Digital. Retrieved from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/how-banks-can-achieve-next-generation-legacy-modernization

Phase 2: Conceptual Data Design

Entities

1. Customer
2. Login
3. Account
4. Savings
5. Checking
6. Transaction
7. Merchant

Relationships Between Entities

- Customer – Login
- Customer – Account
- Account – Savings
- Account – Checking
- Checking – Transaction
- Transaction – Merchant

Cardinality Of Relationships Among Entities

- Customer (mandatory one)  Login (optional one)
- Customer (mandatory one)  Account (mandatory many)
- Account (supertype) – Savings (subtype)
- Account (supertype) – Checking (subtype)
- Checking (mandatory one)  Transaction (optional many)
- Transaction (optional many)  Merchant (mandatory one)

Attributes Of All Entities

Customer
- Customer ID
- Customer Full Name
- Customer Joining Date
- Phone Number
- Email
- Address

Login
- Customer ID
- Username
- Password

Savings
- Account ID
- Interest Rates

Account
- Account ID
- Customer ID
- Account Name
- Account Balance
- Routing Number
- Account Opening Date
- Account Type

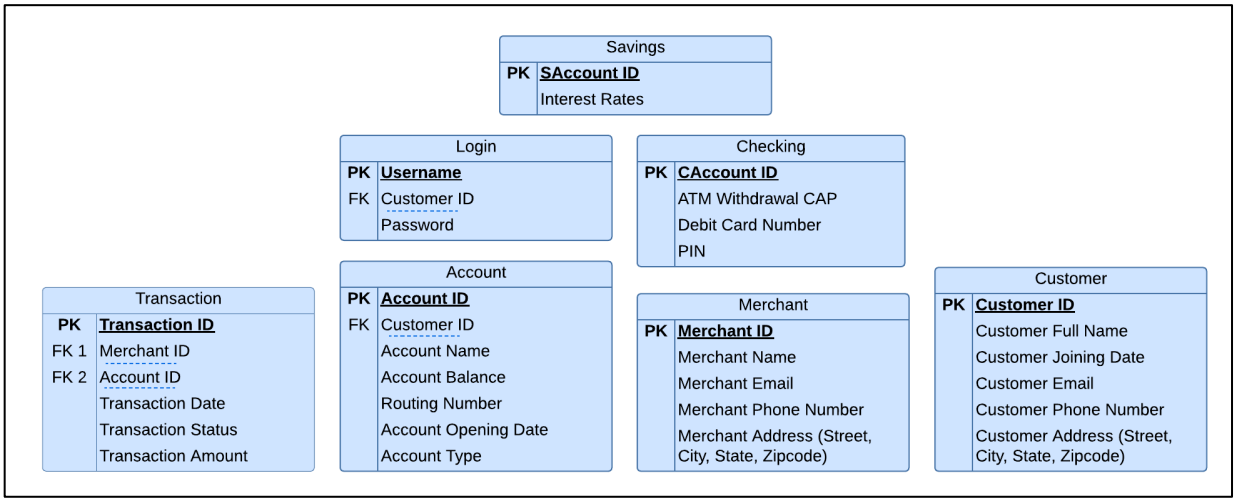
Checking
- Account ID
- ATM Withdrawal CAP
- Debit Card Number
- PIN

Transaction
- Transaction ID
- Merchant ID
- Account ID
- Transaction Date
- Transaction Status
- Transaction Amount

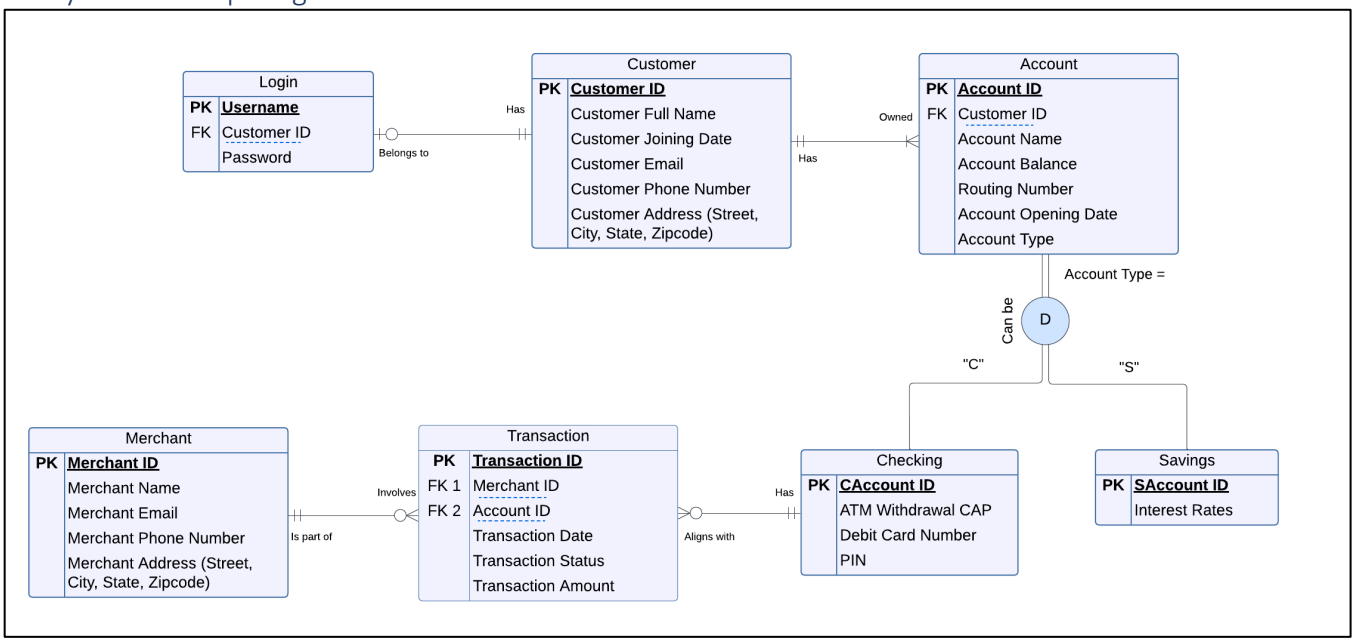
Merchant
- Merchant ID
- Merchant Name
- Merchant Email
- Merchant Phone Number
- Merchant Address

ER-Diagram

Entities And Attributes



Entity Relationship Diagram



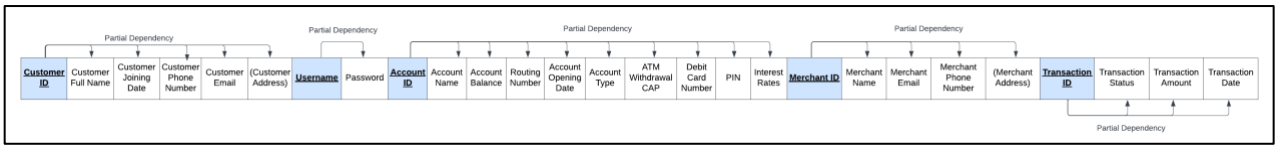
Assumptions:

- Password is dependent on Username, hence is a partial dependency on Username and not Customer ID.
- Swift Bank only has one branch; thus, the routing number remains the same.

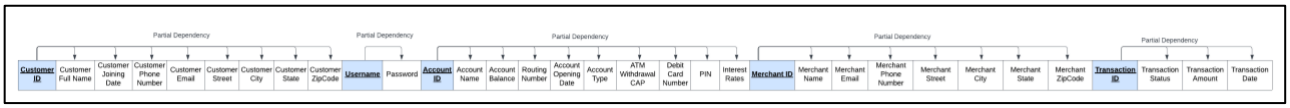
Note: the system could have higher complexity if new entities are created (e.g., 'Deposit'), more attributes become multivalued (e.g., 'Customer Phone'), etc. However, for the purpose of this project, this database system will remain simple.

Phase 3: Logical Data Design

Data Normalization

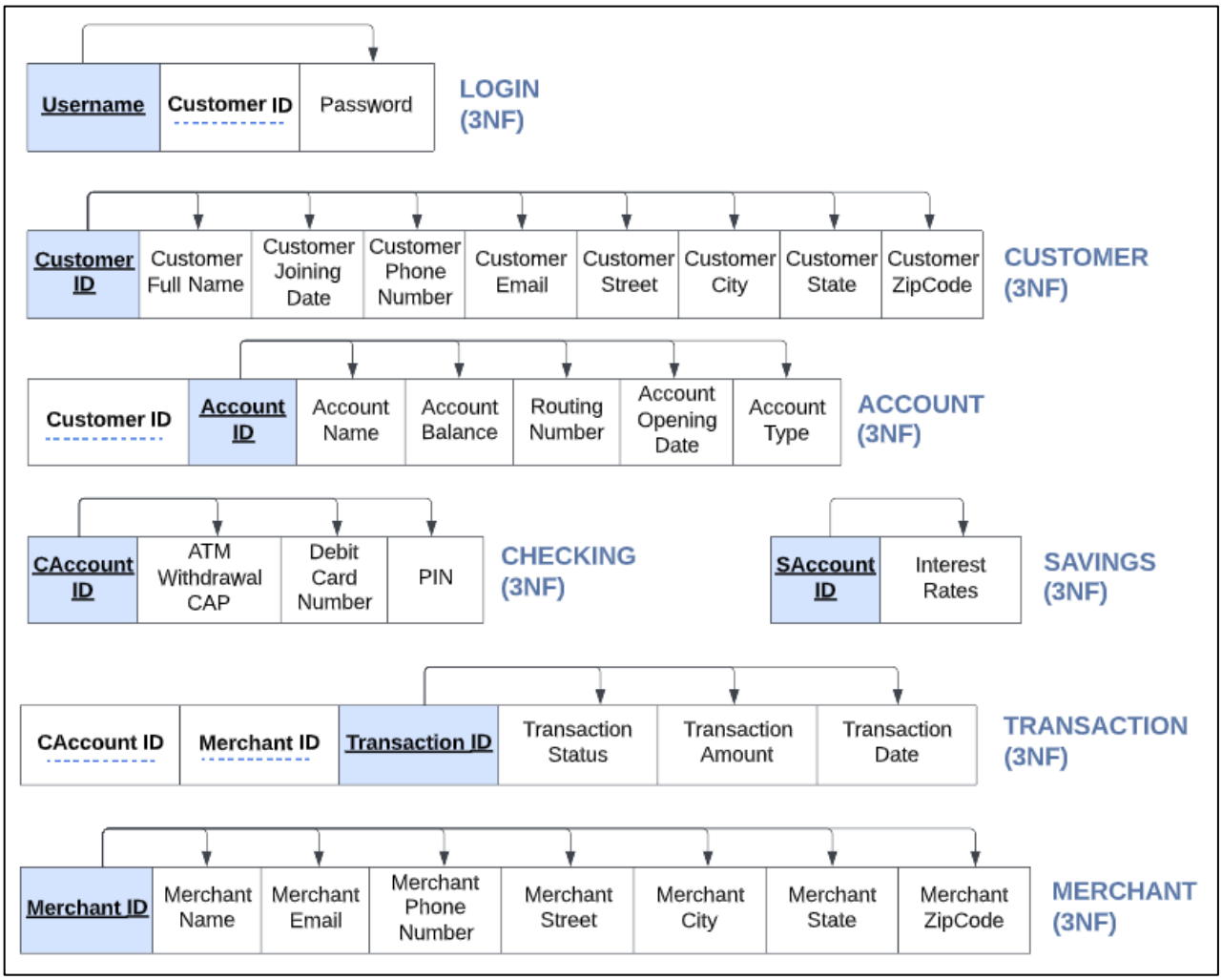


1st Normal Form

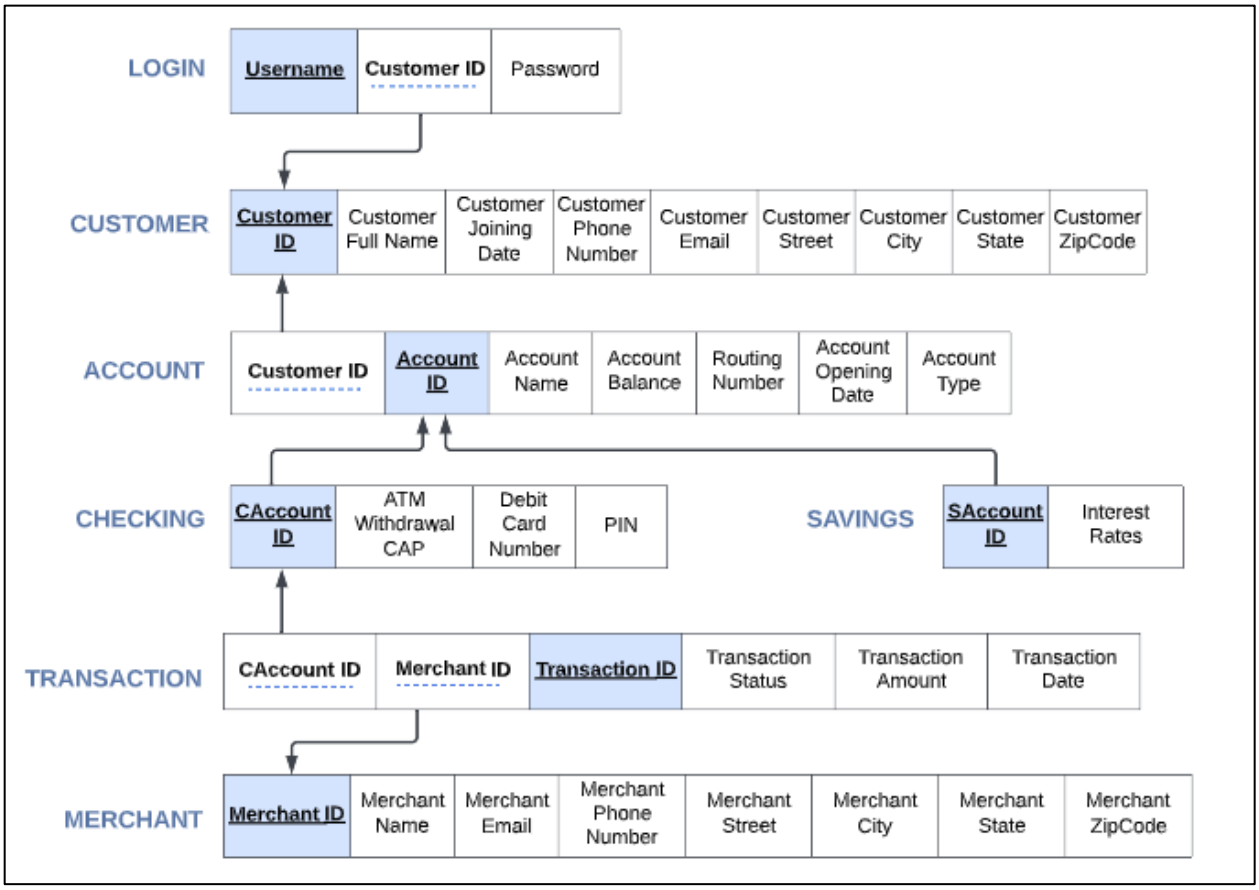


Note: Multivalued attributes resolved and there are no transitive dependencies, therefore jump to 3rd normal form

3rd Normal Form



Relational Schema



Entity Summary Tables

Customer Table

Attribute	Data Type	Constraints
CustomerID	INT	NOT NULL, PK
CustomerFullName	VARCHAR(50)	NOT NULL
CustomerJoiningDate	DATE	NOT NULL
PhoneNumber	VARCHAR(20)	UNIQUE, Check: (___) ___-____)
CustomerEmail	VARCHAR(50)	UNIQUE, Check: @
CustomerStreet	VARCHAR(100)	NOT NULL, Check <>
CustomerCity	VARCHAR(50)	NOT NULL, Check<>
CustomerState	VARCHAR(2)	NOT NULL, Check '__'
CustomerZipcode	VARCHAR(10)	NOT NULL, Check '_____'

Login Table

Attribute	Data Type	Constraints
CUsername	VARCHAR(50)	NOT NULL, PK
CustomerID	INT	NOT NULL, FK
CPassword	VARCHAR(50)	NOT NULL, Length >=8

Accounts Table

Attribute	Data Type	Constraints
AccountID	INT	NOT NULL, PK
CustomerID	INT	NOT NULL, FK
AccountName	VARCHAR(50)	NOT NULL
AccountBalance	DECIMAL(10,2)	NOT NULL
RoutingNumber	VARCHAR(20)	DEFAULT '123456789' NOT NULL
AccountOpeningDate	DATE	NOT NULL
AccountType	VARCHAR(20)	NOT NULL, Check ('Savings, 'Checking')

Checking Table

Attribute	Data Type	Constraints
CAccountID	INT	NOT NULL, PK, FK
ATMWithdrawalCAP	DECIMAL (10,2)	NOT NULL, Check >0
DebitCardNumber	VARCHAR(50)	NOT NULL, UNIQUE
PIN	VARCHAR(10)	NOT NULL, Length=4

Savings Table

Attribute	Data Type	Constraints
SAccountID	INT	NOT NULL, PK, FK
Interest Rate	DECIMAL (4,2)	NOT NULL, >=0

Merchant Table

Attribute	Data Type	Constraints
MerchantID	INT	NOT NULL, PK
MerchantName	VARCHAR(50)	NOT NULL
MerchantEmail	VARCHAR(50)	NOT NULL
MerchantPhone	VARCHAR(20)	NOT NULL, Like (___)___-____
MerchantStreet	VARCHAR(50)	NOT NULL, Like '@' & '.'
MerchantCity	VARCHAR(50)	NOT NULL
MerchantState	VARCHAR(2)	NOT NULL, Length=2
MerchantZipCode	VARCHAR(10)	NOT NULL, Check '_____'

Transactions Table

Attribute	Data Type	Constraints
TransactionID	INT	NOT NULL, PK
MerchantID	INT	NOT NULL, FK
AccountID	INT	NOT NULL, FK
TransactionDate	DATE	NOT NULL
TransactionStatus	VARCHAR(30)	NOT NULL, Check ('Cancelled', 'Successful', 'Disputed', 'Disputed then Resolved', 'Declined')
TransactionAmount	DECIMAL(10,2)	NOT NULL

Phase 4: Physical Data Design

Creation Of Tables

Code:

```
 /*************************IF TABLES EXIST ALREADY*************************/

drop table Customer CASCADE CONSTRAINTS;
drop table Login CASCADE CONSTRAINTS;
drop table Accounts CASCADE CONSTRAINTS;
drop table Checking CASCADE CONSTRAINTS;
drop table Savings CASCADE CONSTRAINTS;
drop table Transactions CASCADE CONSTRAINTS;
drop table Merchant CASCADE CONSTRAINTS;
Commit;


/*CREATION OF TABLES STARTED*/

/*********************CREATING CUSTOMER Table*******************************/

CREATE TABLE Customer (
   CustomerID INT NOT NULL,
   CustomerFullName VARCHAR(50) NOT NULL,
   CustomerJoiningDate DATE NOT NULL,
   PhoneNumber VARCHAR(20) UNIQUE,
   CustomerEmail VARCHAR(50) UNIQUE,
   CustomerStreet VARCHAR(100) NOT NULL,
   CustomerCity VARCHAR(50) NOT NULL,
   CustomerState VARCHAR(2) NOT NULL,
   CustomerZipcode VARCHAR(10) NOT NULL,
   CONSTRAINT PK_Customer PRIMARY KEY (CustomerID),
   CONSTRAINT phone_number_format CHECK (PhoneNumber LIKE '(___) ___-____'),
   CONSTRAINT email_format CHECK (CustomerEmail LIKE '%@%.%'),
   CONSTRAINT zip_format CHECK (CustomerZipcode LIKE '_____'),
   CONSTRAINT state_format CHECK (CustomerState LIKE '__'),
   CONSTRAINT street_not_empty CHECK (CustomerStreet <> ''),
   CONSTRAINT city_not_empty CHECK (CustomerCity <> '')
);

/************************** CREATING LOGIN Table**********************************/

CREATE TABLE Login (
   CUsername VARCHAR(50) NOT NULL PRIMARY KEY,
   CustomerID INT NOT NULL,
```

```
   CPassword VARCHAR(50) NOT NULL,
   CONSTRAINT FK_Login_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
   CONSTRAINT CK_Login_Password CHECK (LENGTH(CPassword) >= 8)
);

/**************************CREATING ACCOUNTS Table*****************************/

CREATE TABLE Accounts (
   AccountID INT NOT NULL PRIMARY KEY,
   CustomerID INT NOT NULL,
   AccountName VARCHAR(50) NOT NULL,
   AccountBalance DECIMAL(10,2) NOT NULL,
   RoutingNumber VARCHAR(20) DEFAULT '123456789' NOT NULL,
   AccountOpeningDate DATE NOT NULL,
   AccountType VARCHAR(20) NOT NULL,
   CONSTRAINT FK_Account_CustomerID FOREIGN KEY (CustomerID) REFERENCES
Customer(CustomerID),
   CONSTRAINT CK_Account_AccountBalance CHECK (AccountBalance >= 0),
   CONSTRAINT CK_Account_AccountType CHECK (AccountType IN ('Savings', 'Checking'))
);

/************************** CREATING CHECKING Table*************************/

CREATE TABLE Checking (
   CAccountID INT NOT NULL PRIMARY KEY,
   ATMWithdrawalCAP DECIMAL(10,2) NOT NULL,
   DebitCardNumber VARCHAR(50) NOT NULL,
   PIN VARCHAR(10) NOT NULL,
   CONSTRAINT FK_Checking_Account FOREIGN KEY (CAccountID) REFERENCES Accounts (AccountID),
   CONSTRAINT CK_Checking_ATMWithdrawalCAP CHECK (ATMWithdrawalCAP > 0),
   CONSTRAINT CK_Checking_PIN CHECK (LENGTH(PIN) = 4),
   CONSTRAINT UQ_Checking_DebitCardNumber UNIQUE (DebitCardNumber)
);


/************************** CREATING SAVINGS Table*******************************/

CREATE TABLE Savings (
   SAccountID INT NOT NULL,
   InterestRate DECIMAL(4,2) NOT NULL,
   CONSTRAINT FK_Savings_AccountID FOREIGN KEY (SAccountID) REFERENCES Accounts(AccountID),
   CONSTRAINT CK_Savings_InterestRate CHECK (InterestRate >= 0),
   CONSTRAINT PK_Saving PRIMARY KEY (SAccountID)
);
```

/*********************** CREATING MERCHANT Table*****************************/**

```
CREATE TABLE Merchant (
   MerchantID INT NOT NULL PRIMARY KEY,
   MerchantName VARCHAR(50) NOT NULL,
   MerchantEmail VARCHAR(50) NOT NULL,
   MerchantPhone VARCHAR(20) NOT NULL,
   MerchantStreet VARCHAR(50) NOT NULL,
   MerchantCity VARCHAR(50) NOT NULL,
   MerchantState VARCHAR(2) NOT NULL,
   MerchantZipcode VARCHAR(10) NOT NULL,
   CONSTRAINT CHK_MerchantState CHECK (LENGTH(MerchantState) = 2),
   CONSTRAINT CHK_MerchantEmail CHECK (MerchantEmail LIKE '%@%.%'),
   CONSTRAINT CHK_MerchantZipcode CHECK (MerchantZipcode LIKE '_____'),
   CONSTRAINT CHK_MerchantPhone CHECK (MerchantPhone LIKE '(___) ___-____')
);
```

/********************** CREATING TRANSACTIONS Table*************************/**

```
CREATE TABLE Transactions (
   TransactionID INT NOT NULL PRIMARY KEY,
   MerchantID INT NOT NULL,
   AccountID INT NOT NULL,
   TransactionDate DATE NOT NULL,
   TransactionStatus VARCHAR(30) NOT NULL CHECK (TransactionStatus IN ('Cancelled', 'Successful',
'Disputed', 'Disputed then Resolved', 'Declined')),
   TransactionAmount DECIMAL(10,2) NOT NULL,
   CONSTRAINT FK_Transaction_MerchantID FOREIGN KEY (MerchantID) REFERENCES
Merchant(MerchantID),
   CONSTRAINT FK_Transaction_AccountID FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
);
```

Data Generation

/*******************INSERTING VALUES IN CUSTOMER TABLE ************************/**

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1001, 'Gauravi Bendre', TO_DATE('2022-01-01', 'YYYY-MM-DD'), '(123) 456-7890', 'john.smith@gmail.com', '123 Main St', 'Anytown', 'CA', '12345');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1002, 'Maria Secaira', TO_DATE('2022-02-15', 'YYYY-MM-DD'), '(234) 567-8901', 'mary.johnson@yahoo.com', '456 Oak Ave', 'Smallville', 'NY', '67890');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1003, 'Sahil Suhag', TO_DATE('2022-03-20', 'YYYY-MM-DD'), '(345) 678-9012', 'bob.smith@hotmail.com', '789 Elm St', 'Bigtown', 'TX', '34567');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1004, 'Maria Fernanda', TO_DATE('2022-04-05', 'YYYY-MM-DD'), '(456) 789-0123', 'jane.doe@gmail.com', '321 Maple St', 'Hometown', 'IL', '45678');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1005, 'Tom Johnson', TO_DATE('2022-05-10', 'YYYY-MM-DD'), '(567) 890-1234', 'tom.johnson@yahoo.com', '654 Pine St', 'Suburbia', 'WA', '89012');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1006, 'Samantha Lee', TO_DATE('2022-06-15', 'YYYY-MM-DD'), '(678) 901-2345', 'samantha.lee@hotmail.com', '987 Cedar St', 'Cityville', 'FL', '23456');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1007, 'David Garcia', TO_DATE('2022-07-20', 'YYYY-MM-DD'), '(789) 012-3456', 'david.garcia@gmail.com', '753 Oak St', 'Townsville', 'AZ', '78901');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1008, 'Emily Davis', TO_DATE('2023-01-29', 'YYYY-MM-DD'), '(890) 123-4567', 'emily.davis@yahoo.com', '246 Pine St', 'Villageville', 'OH', '12345');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber, CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)

```
VALUES (1009, 'Ryan Brown', TO_DATE('2023-02-25', 'YYYY-MM-DD'), '(901) 234-5678',
'ryan.brown@hotmail.com', '864 Maple St', 'Countrytown', 'TN', '67890');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1010, 'Megan Perez', TO_DATE('2023-03-15', 'YYYY-MM-DD'), '(012) 345-6789',
'megan.perez@gmail.com', '975 Cedar St', 'Seaside', 'CA', '34567');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1011, 'Andrew Johnson', TO_DATE('2023-04-20', 'YYYY-MM-DD'), '(123) 496-7890',
'andrew.johnson@gmail.com', '123 Main St', 'Anytown', 'CA', '12345');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1012, 'Sophie Garcia', TO_DATE('2023-05-10', 'YYYY-MM-DD'), '(234) 067-8901',
'sophie.garcia@yahoo.com', '456 Oak Ave', 'Smallville', 'NY', '67890');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1013, 'Alex Nguyen', TO_DATE('2023-06-15', 'YYYY-MM-DD'), '(345) 978-9012',
'alex.nguyen@hotmail.com', '789 Elm St', 'Bigtown', 'TX', '34567');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1014, 'Grace Lee', TO_DATE('2023-07-05', 'YYYY-MM-DD'), '(456) 787-0123',
'grace.lee@gmail.com', '321 Maple St', 'Hometown', 'IL', '45678');

INSERT INTO Customer (CustomerID, CustomerFullName, CustomerJoiningDate, PhoneNumber,
CustomerEmail, CustomerStreet, CustomerCity, CustomerState, CustomerZipcode)
VALUES (1015, 'Daniel Kim', TO_DATE('2023-08-10', 'YYYY-MM-DD'), '(567) 850-1234',
'daniel.kim@yahoo.com', '654 Pine St', 'Suburbia', 'WA', '89012');

/**********INSERTING VALUES IN LOGIN TABLE **********/
INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('gauravibendre', 1001, 'Password123');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('mariasecaira', 1002, 'SecurePassword1');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('sahilsuhag', 1003, 'StrongPassword123');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('mariafernanda', 1004, 'Password4567');

INSERT INTO Login (CUsername, CustomerID, CPassword)
```

```
VALUES ('tomjohnson', 1005, 'MyPasswordIsSecure1');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('samanthalee', 1006, 'Password12345');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('davidgarcia', 1007, 'SecurePassword12');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('emilydavis', 1008, 'StrongPassword1234');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('ryanbrown', 1009, 'Password5678');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('meganperez', 1010, 'MySecurePassword123');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('andrewjohnson', 1011, 'MyPassw0rd456 ');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('sophiegarcia', 1012, 'ElephantsAreCool123');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('alexnguyen', 1013, 'Tr0ub4dor3');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('gracelee', 1014, 'F1shingIsFun99');

INSERT INTO Login (CUsername, CustomerID, CPassword)
VALUES ('danielkim', 1015, '7Horses8Running');
```

/********INSERTING VALUES IN Accounts TABLE **********/**

```
INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (1, 1001, 'Gauravi Bendre - Checking', 5000.00, '123456789', TO_DATE('2022-01-01', 'YYYY-MM-
DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (2, 1006, 'Samantha Lee - Savings', 10000.00, '123456789', TO_DATE('2022-01-01', 'YYYY-MM-
DD'), 'Savings');
```

```
INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (3, 1002, 'Maria Secaira - Checking', 2500.00, '123456789', TO_DATE('2022-02-15', 'YYYY-MM-
DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (4, 1002, 'Maria Secaira - Savings', 15000.00, '123456789', TO_DATE('2022-02-15', 'YYYY-MM-
DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (5, 1003, 'Sahil Suhag - Checking', 1000.00, '123456789', TO_DATE('2022-03-20', 'YYYY-MM-
DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (6, 1007, 'David Garcia - Savings', 5000.00, '123456789', TO_DATE('2022-03-20', 'YYYY-MM-DD'),
'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (7, 1008, 'Emily Davis - Checking', 7000.00, '123456789', TO_DATE('2023-02-05', 'YYYY-MM-DD'),
'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (8, 1004, 'Maria Fernanda - Savings', 12000.00, '123456789', TO_DATE('2022-04-05', 'YYYY-MM-
DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (9, 1009, 'Ryan Brown - Checking', 1500.00, '123456789', TO_DATE('2023-03-27', 'YYYY-MM-
DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (10, 1005, 'Tom Johnson - Savings', 20000.00, '123456789', TO_DATE('2022-05-10', 'YYYY-MM-
DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
VALUES (11, 1010, 'Megan Perez - Checking', 1500.00, '123456789', TO_DATE('2023-03-16', 'YYYY-MM-
DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber,
AccountOpeningDate, AccountType)
```

```
VALUES (12, 1011, 'Andrew Johnson - Savings', 1000.00, '123456789', TO_DATE('2023-04-20', 'YYYY-MM-DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (13, 1011, 'Andrew Johnson - Checking', 5000.00, '123456789', TO_DATE('2023-04-20', 'YYYY-MM-DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (14, 1012, 'Sophie Garcia - Savings', 2500.00, '123456789', TO_DATE('2023-05-10', 'YYYY-MM-DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (15, 1012, 'Sophie Garcia - Checking', 7500.00, '123456789', TO_DATE('2023-05-10', 'YYYY-MM-DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (16, 1013, 'Alex Nguyen - Savings', 5000.00, '123456789', TO_DATE('2023-06-15', 'YYYY-MM-DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (17, 1014, 'Grace Lee - Checking', 10000.00, '123456789', TO_DATE('2023-07-05', 'YYYY-MM-DD'), 'Checking');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (18, 1015, 'Daniel Kim - Savings', 7500.00, '123456789', TO_DATE('2023-08-10', 'YYYY-MM-DD'), 'Savings');

INSERT INTO Accounts (AccountID, CustomerID, AccountName, AccountBalance, RoutingNumber, AccountOpeningDate, AccountType)
VALUES (19, 1015, 'Daniel Kim - Checking', 5000.00, '123456789', TO_DATE('2023-08-10', 'YYYY-MM-DD'), 'Checking');
```

/********INSERTING VALUES IN Checking TABLE **********/**

```
INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (1, 1000.00, '1111222233334444', '1234');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (3, 500.00, '5555666677778888', '4321');
```

```
INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (5, 200.00, '9999000011112222', '7890');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (7, 500.00, '3333444455556666', '5678');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (9, 100.00, '7777888899990000', '2468');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (11, 200.00, '8787222233345444', '1357');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (13, 2000.00, '1299222233345444', '1907');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (15, 8200.00, '9876762233345444', '1567');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (17, 90.00, '1555422233345444', '3357');

INSERT INTO Checking (CAccountID, ATMWithdrawalCAP, DebitCardNumber, PIN)
VALUES (19,3900.00, '1333222233345444', '0987');
```

/********INSERTING VALUES IN Savings TABLE **********/**

```
INSERT INTO Savings (SAccountID, InterestRate)
VALUES (2, 2.50);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (4, 3.00);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (6, 2.00);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (8, 2.75);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (10, 3.25);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (12, 3.95);
```

```
INSERT INTO Savings (SAccountID, InterestRate)
VALUES (14, 2.25);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (16, 1.25);

INSERT INTO Savings (SAccountID, InterestRate)
VALUES (18, 1.55);
```

/********INSERTING VALUES IN MERCHANT TABLE**********/**

```
INSERT INTO Merchant (MerchantID, MerchantName, MerchantEmail, MerchantPhone,
MerchantStreet, MerchantCity, MerchantState, MerchantZipcode)
VALUES (1, 'Harry Potter', 'harrypotter@gmail.com', '(123) 456-7890', '123 Main St', 'Buffalo', 'NY',
'14201');

INSERT INTO Merchant (MerchantID, MerchantName, MerchantEmail, MerchantPhone,
MerchantStreet, MerchantCity, MerchantState, MerchantZipcode)
VALUES (2, 'Jane Smith', 'janesmith@yahoo.com', '(555) 555-5555', '456 Oak Ave', 'Sunnyvale', 'CA',
'94043');

INSERT INTO Merchant (MerchantID, MerchantName, MerchantEmail, MerchantPhone,
MerchantStreet, MerchantCity, MerchantState, MerchantZipcode)
VALUES (3, 'Bob Johnson', 'johnsonbob@hotmail.com', '(999) 999-9999', '789 Elm St', 'Boston', 'MA',
'02129');
```

/********INSERTING VALUES IN Transaction TABLE **********/**

```
INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (1, 1, 1, TO_DATE('2023-01-23', 'YYYY-MM-DD'), 'Successful', 20.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (2, 1, 1, TO_DATE('2023-03-02', 'YYYY-MM-DD'), 'Declined', 50.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (3, 1, 3, TO_DATE('2023-01-28', 'YYYY-MM-DD'), 'Disputed', 10.00);
```

```
INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (4, 1, 5, TO_DATE('2023-02-04', 'YYYY-MM-DD'), 'Successful', 100.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (5, 3, 7, TO_DATE('2023-01-05', 'YYYY-MM-DD'), 'Cancelled', 30.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (13, 1, 7,TO_DATE('2023-03-11', 'YYYY-MM-DD'), 'Successful', 45.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (14, 1, 9, TO_DATE('2023-02-12', 'YYYY-MM-DD'), 'Disputed then Resolved', 75.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (15, 1, 11, TO_DATE('2023-03-12', 'YYYY-MM-DD'), 'Cancelled', 120.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (16, 1, 5, TO_DATE('2023-03-13', 'YYYY-MM-DD'), 'Successful', 25.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (17, 2, 1, TO_DATE('2023-03-13', 'YYYY-MM-DD'), 'Declined', 85.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (18, 1, 3, TO_DATE('2023-03-13', 'YYYY-MM-DD'), 'Disputed', 60.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (19, 2, 11, TO_DATE('2023-03-14', 'YYYY-MM-DD'), 'Successful', 40.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (20, 3, 9, TO_DATE('2023-03-14', 'YYYY-MM-DD'), 'Cancelled', 90.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (21, 1, 5, TO_DATE('2023-03-14', 'YYYY-MM-DD'), 'Successful', 55.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
```

```
VALUES (22, 1, 7, TO_DATE('2023-03-14', 'YYYY-MM-DD'), 'Disputed then Resolved', 100.00);

 INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (23, 1, 1, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 35.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (24, 1, 3, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Disputed', 70.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (25, 1, 7, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Cancelled', 25.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (26, 1, 9, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 80.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (27, 1, 5, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Declined', 90.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (28, 1, 11, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 65.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
 VALUES (29, 1, 7, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Disputed then Resolved', 55.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (30, 1, 3, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Cancelled', 15.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (31, 1, 5, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 30.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (32, 2, 9, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Disputed', 95.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus,
TransactionAmount)
VALUES (33, 1, 13, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 24.99);
```

```
INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (34, 2, 15, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 50.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (35, 1, 17, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Declined', 10.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (36, 1, 19, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 100.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (37, 1, 13, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 75.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (38, 1, 15, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Disputed', 200.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (39, 1, 17, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 12.50);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (40, 2, 19, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 50.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (41, 1, 13, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Disputed then Resolved', 150.00);

INSERT INTO Transactions (TransactionID, MerchantID, AccountID, TransactionDate, TransactionStatus, TransactionAmount)
VALUES (42, 1, 15, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 'Successful', 8.99);
```

Entity Tables

/**********VIEWING DATA FROM THE TABLES **********/

1. **LOGIN TABLE:**

Code:

*Select * from Login;*

Output:

	CUSERNAME	CUSTOMERID	CPASSWORD
1	gauravibendre	1001	Password123
2	mariasecaira	1002	SecurePassword1
3	sahilsuhag	1003	StrongPassword123
4	mariafernanda	1004	Password4567
5	tomjohnson	1005	MyPasswordIsSecure1
6	samanthalee	1006	Password12345
7	davidgarcia	1007	SecurePassword12
8	emilydavis	1008	StrongPassword1234
9	ryanbrown	1009	Password5678
10	meganperez	1010	MySecurePassword123
11	andrewjohnson	1011	MyPassw0rd456
12	sophiegarcia	1012	ElephantsAreCool123
13	alexnguyen	1013	Tr0ub4dor3
14	gracelee	1014	FlshingIsFun99
15	danielkim	1015	7Horses8Running

2. CUSTOMER TABLE:

Code:

*Select * from Customer;*

Output:

	CUSTOMERID	CUSTOMERFULLNAME	CUSTOMERJOININGDATE	PHONENUMBER	CUSTOMEREMAIL	CUSTOMERSTREET	CUSTOMERCITY	CUSTOMERSTATE	CUSTOMERZIPCODE
1	1001	Gauravi Bendre	01-JAN-22	(123) 456-7890	john.smith@gmail.com	123 Main St	Anytown	CA	12345
2	1002	Maria Secaira	15-FEB-22	(234) 567-8901	mary.johnson@yahoo.com	456 Oak Ave	Smallville	NY	67890
3	1003	Sahil Suhag	20-MAR-22	(345) 678-9012	bob.smith@hotmail.com	789 Elm St	Bigtown	TX	34567
4	1004	Maria Fernanda	05-APR-22	(456) 789-0123	jane.doe@gmail.com	321 Maple St	Hometown	IL	45678
5	1005	Tom Johnson	10-MAY-22	(567) 890-1234	tom.johnson@yahoo.com	654 Pine St	Suburbia	WA	89012
6	1006	Samantha Lee	15-JUN-22	(678) 901-2345	samantha.lee@hotmail.com	987 Cedar St	Cityville	FL	23456
7	1007	David Garcia	20-JUL-22	(789) 012-3456	david.garcia@gmail.com	753 Oak St	Townsville	AZ	78901
8	1008	Emily Davis	29-JAN-23	(890) 123-4567	emily.davis@yahoo.com	246 Pine St	Villageville	OH	12345
9	1009	Ryan Brown	25-FEB-23	(901) 234-5678	ryan.brown@hotmail.com	864 Maple St	Countrytown	TN	67890
10	1010	Megan Perez	15-MAR-23	(012) 345-6789	megan.perez@gmail.com	975 Cedar St	Seaside	CA	34567
11	1011	Andrew Johnson	20-APR-23	(123) 496-7890	andrew.johnson@gmail.com	123 Main St	Anytown	CA	12345
12	1012	Sophie Garcia	10-MAY-23	(234) 067-8901	sophie.garcia@yahoo.com	456 Oak Ave	Smallville	NY	67890
13	1013	Alex Nguyen	15-JUN-23	(345) 978-9012	alex.nguyen@hotmail.com	789 Elm St	Bigtown	TX	34567
14	1014	Grace Lee	05-JUL-23	(456) 787-0123	grace.lee@gmail.com	321 Maple St	Hometown	IL	45678
15	1015	Daniel Kim	10-AUG-23	(567) 850-1234	daniel.kim@yahoo.com	654 Pine St	Suburbia	WA	89012

3. ACCOUNTS TABLE:

Code:

*Select * from Accounts;*

Output:

	ACCOUNTID	CUSTOMERID	ACCOUNTNAME	ACCOUNTBALANCE	ROUTINGNUMBER	ACCOUNTOPENINGDATE	ACCOUNTTYPE
1	1	1001	Gauravi Bendre - Checking	5000	123456789	01-JAN-22	Checking
2	2	1006	Samantha Lee - Savings	10000	123456789	01-JAN-22	Savings
3	3	1002	Maria Secaira - Checking	2500	123456789	15-FEB-22	Checking
4	4	1002	Maria Secaira - Savings	15000	123456789	15-FEB-22	Savings
5	5	1003	Sahil Suhag - Checking	1000	123456789	20-MAR-22	Checking
6	6	1007	David Garcia - Savings	5000	123456789	20-MAR-22	Savings
7	7	1008	Emily Davis - Checking	7000	123456789	05-FEB-23	Checking
8	8	1004	Maria Fernanda - Savings	12000	123456789	05-APR-22	Savings
9	9	1009	Ryan Brown - Checking	1500	123456789	27-MAR-23	Checking
10	10	1005	Tom Johnson - Savings	20000	123456789	10-MAY-22	Savings
11	11	1010	Megan Perez - Checking	1500	123456789	16-MAR-23	Checking
12	12	1011	Andrew Johnson - Savings	1000	123456789	20-APR-23	Savings
13	13	1011	Andrew Johnson - Checking	5000	123456789	20-APR-23	Checking
14	14	1012	Sophie Garcia - Savings	2500	123456789	10-MAY-23	Savings
15	15	1012	Sophie Garcia - Checking	7500	123456789	10-MAY-23	Checking
16	16	1013	Alex Nguyen - Savings	5000	123456789	15-JUN-23	Savings
17	17	1014	Grace Lee - Checking	10000	123456789	05-JUL-23	Checking
18	18	1015	Daniel Kim - Savings	7500	123456789	10-AUG-23	Savings
19	19	1015	Daniel Kim - Checking	5000	123456789	10-AUG-23	Checking

4. **SAVINGS TABLE:**

Code:

*Select * from Savings;*

Output:

	SACCOUNTID	INTERESTRATE
1	2	2.5
2	4	3
3	6	2
4	8	2.75
5	10	3.25
6	12	3.95
7	14	2.25
8	16	1.25
9	18	1.55

5. **CHECKING TABLE:**

Code:

*Select * from Checking;*

Output:

	CACCOUNTID	ATMWITHDRAWALCAP	DEBITCARDNUMBER	PIN
1	1	1000	1111222233334444	1234
2	3	500	5555666677778888	4321
3	5	200	9999000011112222	7890
4	7	500	3333444455556666	5678
5	9	100	7777888899990000	2468
6	11	200	8787222233345444	1357
7	13	2000	1299222233345444	1907
8	15	8200	9876762233345444	1567
9	17	90	1555422233345444	3357
10	19	3900	1333222233345444	0987

6. TRANSACTIONS TABLE:

Code:

*Select * from Transactions;*

Output:

	TRANSACTIONID	MERCHANTID	ACCOUNTID	TRANSACTIONDATE	TRANSACTIONSTATUS	TRANSACTIONAMOUNT
1	1	1	1	23-JAN-23	Successful	20
2	2	1	1	02-MAR-23	Declined	50
3	3	1	3	28-JAN-23	Disputed	10
4	4	2	5	04-FEB-23	Successful	100
5	5	3	7	05-JAN-23	Cancelled	30
6	13	3	7	11-MAR-23	Successful	45
7	14	2	9	12-FEB-23	Disputed then Resolved	75
8	15	1	11	12-MAR-23	Cancelled	120
9	16	3	5	13-MAR-23	Successful	25
10	17	2	1	13-MAR-23	Declined	85
11	18	1	3	13-MAR-23	Disputed	60
12	19	2	11	14-MAR-23	Successful	40
13	20	3	9	14-MAR-23	Cancelled	90
14	21	1	5	14-MAR-23	Successful	55
15	22	2	7	14-MAR-23	Disputed then Resolved	100
16	23	3	1	15-MAR-23	Successful	35
17	24	2	3	15-MAR-23	Disputed	70
18	25	1	7	15-MAR-23	Cancelled	25
19	26	3	9	15-MAR-23	Successful	80
20	27	2	5	15-MAR-23	Declined	90
21	28	1	11	15-MAR-23	Successful	65
22	29	2	7	15-MAR-23	Disputed then Resolved	55
23	30	3	3	15-MAR-23	Cancelled	15
24	31	1	5	15-MAR-23	Successful	30
25	32	2	9	15-MAR-23	Disputed	95
26	33	1	13	15-MAR-23	Successful	24.99
27	34	2	15	15-MAR-23	Successful	50
28	35	3	17	15-MAR-23	Declined	10

7. MERCHANT TABLE:

Code:

*Select * from Merchant;*

	TRANSACTIONID	MERCHANTID	ACCOUNTID	TRANSACTIONDATE	TRANSACTIONSTATUS	TRANSACTIONAMOUNT
28	35	3	17	15-MAR-23	Declined	10
29	36	2	19	15-MAR-23	Successful	100
30	37	1	13	15-MAR-23	Successful	75
31	38	3	15	15-MAR-23	Disputed	200
32	39	1	17	15-MAR-23	Successful	12.5
33	40	2	19	15-MAR-23	Successful	50
34	41	3	13	15-MAR-23	Disputed then Resolved	150
35	42	1	15	15-MAR-23	Successful	8.99

Output:

	MERCHANTID	MERCHANTNAME	MERCHANTEMAIL	MERCHANTPHONE	MERCHANTSTREET	MERCHANTCITY	MERCHANTSTATE	MERCHANTZIPCODE
1	1	Harry Potter	harrypotter@gmail.com	(123) 456-7890	123 Main St	Buffalo	NY	14201
2	2	Jane Smith	janesmith@yahoo.com	(555) 555-5555	456 Oak Ave	Sunnyvale	CA	94043
3	3	Bob Johnson	johnsonbob@hotmail.com	(999) 999-9999	789 Elm St	Boston	MA	02129

Phase 5: Application and Learnings

Database Query Examples

/***********************Business Questions *************************/**

1. Retrieve and analyze the **number of records per transaction status** in the bank database management system. This helps businesses identify how many of the transactions were successful and how many belong to the other categories. This will also allow the business to analyze transaction performance and identify areas for improvement.

   Code:

   ```
   SELECT TRANSACTIONSTATUS as "Transaction Status",
   COUNT(*) AS "Transactions Per Status"
   FROM TRANSACTIONS
   GROUP BY TRANSACTIONSTATUS;
   ```

   Output:

	TRANSACTIONSTATUS	Transactions Per Status
1	Successful	17
2	Declined	4
3	Disputed	5
4	Cancelled	5
5	Disputed then Resolved	4

2. **Determine the number of transactions that each merchant is involved in.** To answer this, the Transactions table will be used by using the count function, then joined with the Merchant Table to find the Merchant Name. Knowing the answer to this question will can help the bank understand and identify any potential fraudulent activities and take appropriate measure to prevent them, the bank can also better monitor merchant activities and ensure compliance with regulatory requirements.

   Code:

   ```
   SELECT T.MERCHANTID AS "Merchant ID",
   M.MERCHANTNAME AS "Merchant Name",
   COUNT(T.MerchantID) AS "Transactions per merchant"
   FROM TRANSACTIONS T
   INNER JOIN MERCHANT M
   ON M.MERCHANTID = T.MERCHANTID
   GROUP BY T.MERCHANTID, M.MERCHANTNAME
   ORDER BY COUNT(T.MerchantID)DESC;
   ```

Output:

	Merchant ID	Merchant Name	Transactions per merchant
1	1	Harry Potter	28
2	2	Jane Smith	5
3	3	Bob Johnson	2

3. **Identifying customers who possess both savings and checking accounts with the bank.** This question can provide insights into customer behavior and preferences, allowing the bank to better understand the needs of its customers and tailor its services and products to meet those needs. As well, it will enable the bank to segment its customer base more effectively and develop targeted marketing campaigns to reach specific customer groups with the right message.

Code:

```
SELECT DISTINCT A1.CUSTOMERID AS "Customer ID" ,
C.CUSTOMERFULLNAME AS "Customer Name"
FROM ACCOUNTS A1
INNER JOIN ACCOUNTS A2 ON A1.CUSTOMERID = A2.CUSTOMERID
INNER JOIN CUSTOMER C ON A1.CUSTOMERID = C.CUSTOMERID
WHERE A1.ACCOUNTTYPE = 'Savings'
AND A2.ACCOUNTTYPE = 'Checking';
```

Output:

	Customer ID	Customer Name
1	1002	Maria Secaira
2	1011	Andrew Johnson
3	1012	Sophie Garcia
4	1015	Daniel Kim

4. **Identifying customer with the highest cancelled transactions.** This information can be used by the bank to prioritize customer service efforts for those customers who may be experiencing the most issues with cancelled transactions as these cancelled transactions can help the bank deepen the analysis and identify common reasons for cancelled transactions, such as technical difficulties or confusing processes, and work towards resolving them.

Code:

```
SELECT
c.CUSTOMERFULLNAME as "Customer Name",
COUNT(t.TRANSACTIONID) AS "No. of Cancelled Transactions"
FROM CUSTOMER c
INNER JOIN ACCOUNTS a ON c.CUSTOMERID = a.CUSTOMERID
INNER JOIN TRANSACTIONS t ON a.ACCOUNTID = t.ACCOUNTID

WHERE t.TRANSACTIONSTATUS = 'Cancelled'
GROUP BY c.CUSTOMERFULLNAME
HAVING COUNT(t.TRANSACTIONID) =
(
        SELECT
        MAX(cancelled_transactions)
        FROM (
                SELECT
                COUNT(t2.TRANSACTIONID) AS cancelled_transactions
                FROM
                TRANSACTIONS t2
                INNER JOIN ACCOUNTS a2 ON t2.ACCOUNTID = a2.ACCOUNTID
                INNER JOIN CUSTOMER c2 ON a2.CUSTOMERID = c2.CUSTOMERID
                WHERE t2.TRANSACTIONSTATUS = 'Cancelled'
                GROUP BY c2.CUSTOMERFULLNAME
              )
  );
```

Output:

	Customer Name	No. of Cancelled Transactions
1	Emily Davis	2

Learning Achievements

Throughout this project on developing a bank management system for the Swift Bank startup, we have gained an extensive knowledge base and skillset in database management and also the banking and financial services industry. There are several important lessons about database management's role in the banking and financial services domain such as the fact that a reliable and well-designed bank database management system is crucial for financial institutions to accurately and efficiently store, manage, and process sensitive financial information, as the lack of it can lead to inaccuracies, poor customer service, and reputational damage, which can ultimately result in a loss of business. Another essential lesson in the financial context is the need for data accuracy and consistency which was a challenge when developing the physical database, as these two are critical to prevent errors or inconsistencies that can potentially lead to financial losses, regulatory issues, and reputational damage.

Moreover, another key learning achievement from this project was the ability to create a comprehensive conceptual data design, involving entities, relationships, and cardinalities, resulting in the development of an ERD. Additionally, the logical data design phase provided valuable lessons on converting ERDs into relational schemas, and the process of data normalization till 3NF. Mapping subtypes and supertypes further strengthened our understanding of these complex relationships. Lastly, the physical data design phase, utilizing SQL, was also an exciting opportunity to apply our theoretical knowledge to practical application and demonstrate the potential benefits of utilizing this system for real-world business applications.

Overall, this project has provided us with a comprehensive understanding of the entire database development process, from conceptual design to physical implementation. Through this project, we have also developed critical thinking skills, problem-solving skills, and analytical skills, all of which are essential in the field of database management. Last but not least, we have also gained a deeper appreciation for the critical role of database management in ensuring the success of financial institutions and their customers.