1. \*User Logins\*: Stores user login information.
2. \*UserSecurityQuestions\*: Stores security questions for user.

3. \*AccountType\*: Defines types of accounts like savings or checking.

4. \*SavingsInterestRates\*: Stores interest rates for savings accounts.

5. \*AccountStatusType\*: Defines status types for accounts like closed or active.

6. \*FailedTransactionErrorType\*: Defines types of errors for failed transactions.

7. \*LoginErrorLog\*: Logs errors that occur during login attempts.

8. \*Employee\*: Stores employee information.

9. \*TransactionType\*: Defines types of transactions.

10. \*FailedTransactionLog\*: Logs failed transactions.

11. \*UserSecurityAnswers\*: Stores user security question answers.

12. \*Account\*: Stores account information.

13. \*LoginAccount\*: Maps user logins to accounts.

14. \*Customer details\*: Stores customer information.

15. \*CustomerAccount\*: Maps customers to accounts.

16. \*TransactionLog\*: Logs transactions.

17. \*OverDraftLog\*: Logs overdraft occurrences.

The INSERT INTO statements populate these tables with sample data. Each table represents different aspects of the banking system, such as users, accounts, transactions, and errors.

This code segment is creating a database schema with multiple tables and then populating those tables with sample data. Let's break it down:

1. \*UserLogins Table\*: It stores user login information, including a unique identifier (UserLoginID), login name (UserLogin), and password (UserPassword). The constraint pk\_UL\_UserLoginID defines the primary key.

2. \*UserSecurityQuestions Table\*: Stores security questions for user authentication. It has a unique identifier for each question (UserSecurityQuestionID) and the actual question (UserSecurityQuestion). The primary key is pk\_USQ\_UserSecurityQuestionID.

3. \*AccountType Table\*: Holds different types of accounts with a unique identifier (AccountTypeID) and a description (AccountTypeDescription). pk\_AT\_AccountTypeID is the primary key.

4. \*SavingsInterestRates Table\*: Stores interest rates for savings accounts. It includes an identifier for each rate (InterestSavingRatesID), the rate value (InterestRatesValue), and a description (InterestRatesDescription). The primary key is pk\_SIR\_InterestSavingRatesID.

5. \*AccountStatusType Table\*: Defines various statuses for accounts. It has an identifier for each status (AccountStatusTypeID) and a description (AccountStatusTypeDescription). The primary key is pk\_AST\_AccountStatusTypeID.

6. \*FailedTransactionErrorType Table\*: Lists types of errors for failed transactions. It includes an identifier for each error type (FailedTransactionErrorTypeID) and a description (FailedTransactionErrorTypeDescription). The primary key is pk\_FTET\_FailedTransactionErrorTypeID.

7. \*LoginErrorLog Table\*: Records errors during login attempts. It stores an error identifier (ErrorLogID), the time of the error (ErrorTime), and XML data related to the failed transaction (FailedTransactionXML). The primary key is pk\_LEL\_ErrorLogID.

8. \*Employee Table\*: Stores employee information, including an identifier (EmployeeID), first name (EmployeeFirstName), middle initial (EmployeeMiddleInitial), last name (EmployeeLastName), and a flag indicating if the employee is a manager (EmployeeisManager). The primary key is pk\_E\_EmployeeID.

9. \*TransactionType Table\*: Defines types of transactions. It includes an identifier for each type (TransactionTypeID), a name (TransactionTypeName), a description (TransactionTypeDescription), and a fee amount (TransactionFeeAmount). The primary key is pk\_TT\_TransactionTypeID.

10. \*FailedTransactionLog Table\*: Logs failed transactions. It records the identifier of each failed transaction (FailedTransactionID), the type of error (FailedTransactionErrorTypeID), the time of the error (FailedTransactionErrorTime), and XML data related to the error (FailedTransactionErrorXML). The primary key is pk\_FTL\_FailedTransactionID, and it also has a foreign key constraint fk\_FTET\_FailedTransactionErrorTypeID referencing the FailedTransactionErrorType table.

WHAT IS DBMS?

Microsoft SQL Server is a relational database management system. As a database server that stores and retrieves data as requested by other software applications on the same computer or a remote computer using the client-server model. Microsoft provides APIs to access SQL Server over the internet as a web service.

Why is it called SQL Server?

SQL was invented in the 1970s based on the relational data model. It was initially known as the structured English query language (SEQUEL).