Podemos implementar la lógica del cliente y el servidor para que se adecue a esto:

**Tablas:**

**TABLA Cliente**

|  |  |  |
| --- | --- | --- |
| CuentaID | Int | Autogenerado |
| UserID | Nchar(16) | El usuario a usar en el LOG IN |
| PIN | Int | Integer 4 digitos max |
| Nombre | Nchar(16) |  |
| Apellido | Nchar(16) |  |

**\*PIN2 lo volamos porque no lo usamos en ningún lado, es redúndate y ocupa espacio al pedo.**

**TABLA Cuenta**

|  |  |  |
| --- | --- | --- |
| CuentaID | Int | Autogenerado |
| Saldo | Real | Equivalente a float .net, DEFAULT = 0 |
| Descubierto | Real | DEFAULT = 1000 |
| ID\_Cliente | Int | Secondary key |
| ID\_CuentaTipo | Char(3) | Secondary key |

**TABLA CuentaTipo**

|  |  |  |
| --- | --- | --- |
| CuentaTipoID | Char(3) | CA=’Caja de Ahorro’ y CC=’Cuenta Corriente’ |
| Descripcion | Nchar(18) |  |

**TABLA TransactionLog**

|  |  |  |
| --- | --- | --- |
| TransactionID | Int | Autogenerado |
| Type | Nchar(16) | LOGIN, DEPOSITO, EXTRAER, TRANSFERENCIA…. |
| ClienteID | Int |  |
| CtaOrigen | Nchar(16) |  |
| CtaOrigenTipo | Char(2) | CA o CC |
| CtaDestino | Nchar(16) |  |
| CtaDestinoTipo | Char(2) | CA o CC |
| Monto | Real |  |
| TransactionCommit | Bit |  |
| TransactionText | Nchar(256) |  |

**Store Procedures:**

***Cliente\_LOG IN:***

ENTRADA:

* *@UserID* - Nchar(16): Identificación del cliente para el acceso al cliente
* *@PIN* - Int: numero de 4 dígitos.

SALIDA:

* *@TransactionID* – Int: Devuelve el número de ID de la transacción insertada en la tabla TransactionLog.
* *@RETURN\_VALUE* – Int: Retorno del Store Procedure
  + ***1***: El LOG IN fue Satisfactorio
  + ***0***: El PIN ingresado no pertenece a la cuenta ingresada.
  + ***-1***: El usuario ingresado no pertenece a ningún cliente en la BD.

***Cuenta\_GetSaldo:***

ENTRADA:

* *@UserID* - Nchar(16): Identificación del cliente utilizada en el log in.

SALIDA:

* *@TransactionID* – Int: Devuelve el número de ID de la transacción insertada en la tabla TransactionLog.
* *@SaldoActual* – Real: Devuelve la suma de las cuentas pertenecientes al cliente @UserID.
* *@RETURN\_VALUE* – Int: Retorno del Store Procedure
  + ***1***: La operación se realizó con éxito.
  + ***-1***: El usuario ingresado no pertenece a ningún cliente en la BD.

***Cuenta \_Deposito:***

ENTRADA:

* *@UserID* - Nchar(16): Identificación del cliente para el acceso al cliente
* *@Deposito –* Real: Monto a depositar.

SALIDA:

* *@TransactionID* – Int: Devuelve el número de ID de la transacción insertada en la tabla TransactionLog.
* *@RETURN\_VALUE* – Int: Retorno del Store Procedure
  + ***1***: La operación se realizó con éxito.
  + ***-1***: El usuario ingresado no pertenece a ningún cliente en la BD.

CASOS DE NEGOCIO:

1. @UserID no existe en la base de datos:
   1. @RETURN\_VALUE = -1
2. @UserID posee solo una Caja de Ahorro:
   1. Sin importar el saldo actual de la caja de ahorro se suma @Deposito.
   2. @RETURN\_VALUE = 1
3. @UserID posee solo una Cuenta Corriente:
   1. Sin importar el saldo actual de la cuenta corriente se suma @Deposito.
   2. @RETURN\_VALUE = 1
4. @UserID posee una Caja de Ahorro y Cuenta Corriente:
   1. Saldo en Cuenta Corriente < 0 (negativo):
      1. 1 paso: Se suma @Deposito a la cuenta corriente.
      2. 2 paso: Si el saldo de la cuenta corriente es > 0,

Saldo CA = Saldo CC

Saldo CC = 0

@RETURN\_VALUE = 1

End Si

* 1. Saldo en Cuenta Corriente = 0 implica Caja de Ahorro >= 0:
     1. 1 paso: Se suma @Deposito a la caja de ahorro.
     2. @RETURN\_VALUE = -1

***Cuenta \_Extraer:***

ENTRADA:

* *@UserID* - Nchar(16): Identificación del cliente para el acceso al cliente
* *@Extraer –* Real: Monto a extraer.

SALIDA:

* *@TransactionID* – Int: Devuelve el número de ID de la transacción insertada en la tabla TransactionLog.
* *@RETURN\_VALUE* – Int: Retorno del Store Procedure
  + ***1***: La operación fue realizada con éxito.
  + ***0***: El sado es insuficiente.
  + ***-1***: El usuario ingresado no pertenece a ningún cliente en la BD.

CASOS DE NEGOCIO:

1. @UserID no existe en la base de datos:
   1. @RETURN\_VALUE = -1
2. @UserID posee solo una Caja de Ahorro:
   1. Si el saldo de la Caja de Ahorro > @Extraer
      1. 1 paso: Efectuo la Extracion, resto @Extraer del saldo de la caja de ahorro.
      2. @RETURN\_VALUE = 1
   2. Si el saldo de la Caja de Ahorro < @Extraer
      1. @RETURN\_VALUE = 0
3. @UserID posee solo una Cuenta Corriente:
   1. Si el saldo de la Cta Corriente + Descubierto de la Cta Corriente > @Extraer
      1. 1 paso: Efectuó la Extracción, resto @Extraer del saldo de la cuenta corriente.
      2. @RETURN\_VALUE = 1
   2. Si el saldo de la Cuenta Corriente + Descubierto de la Cta Corriente < @Extraer
      1. @RETURN\_VALUE = 0
4. @UserID posee una Caja de Ahorro y Cuenta Corriente:
   1. Saldo en Caja de Ahorro > @Extraer
      1. 1 paso: Efectuó la Extracción, resto @Extraer del saldo de la caja de ahorro.
      2. @RETURN\_VALUE = 1
   2. Saldo en Caja de Ahorro < @Extraer && Caja de Ahorro + Cta Corriente + Descubierto > @Extraer
      1. 1 paso: Saldo Cta Corriente = Saldo Caja de Ahorro - @Extraer.
      2. 2 paso: Saldo Caja de Ahorro = 0
      3. @RETURN\_VALUE = 1
   3. Saldo en Caja de Ahorro + Descubierto < @Extraer
      1. @RETURN\_VALUE = 0