Advanced DataBases, First Delivery

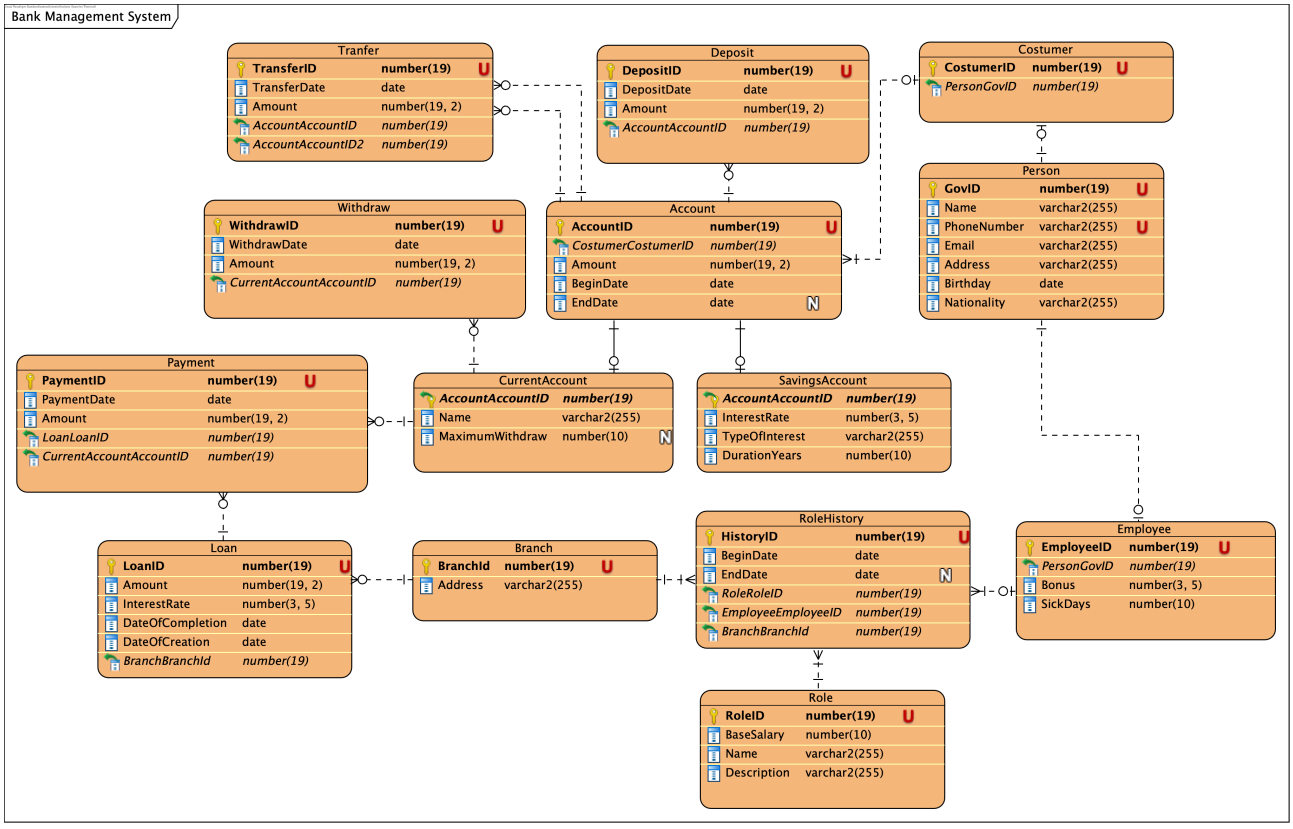
Bank Management System

Samuel Lúcio Vicente, 251720 Daniel Silva, 251702 Jorge Espanhol, XXXXXX

Politechnika Wrocławska
C4 Group
October 17, 2019

Short Description

ERD



Schema

```
1  CREATE TABLE Person (
2    GovID number(19) GENERATED AS IDENTITY,
3    Name varchar2(255) NOT NULL,
4    PhoneNumber varchar2(255) NOT NULL UNIQUE,
5    Email varchar2(255) NOT NULL,
6    Address varchar2(255) NOT NULL,
```

```
7    Birthday date NOT NULL,
8    Nationality varchar2(255) NOT NULL,
9    PRIMARY KEY (GovID));
10
11   CREATE TABLE Costumer (
12     CostumerID number(19) GENERATED AS IDENTITY,
13     PersonGovID number(19) NOT NULL,
14     PRIMARY KEY (CostumerID));
15
16   CREATE TABLE Employee (
17     EmployeeID number(19) GENERATED AS IDENTITY,
18     PersonGovID number(19) NOT NULL,
19     Bonus number(3, 5) NOT NULL CHECK(Bonus>=0),
20     SickDays number(10) NOT NULL CHECK(SickDays<10),
21     PRIMARY KEY (EmployeeID));
22
23   CREATE TABLE Account (
24     AccountID number(19) GENERATED AS IDENTITY,
25     CostumerCostumerID number(19) NOT NULL,
26     Amount number(19, 2) NOT NULL CHECK(Amount>=0),
27     BeginDate date NOT NULL,
28     EndDate date,
29     PRIMARY KEY (AccountID));
30
31   CREATE TABLE Branch (
32     BranchId number(19) GENERATED AS IDENTITY,
33     Address varchar2(255) NOT NULL,
34     PRIMARY KEY (BranchId));
35
36   CREATE TABLE RoleHistory (
37     HistoryID number(19) GENERATED AS IDENTITY,
38     BeginDate date NOT NULL,
39     EndDate date,
40     RoleRoleID number(19) NOT NULL,
41     EmployeeEmployeeID number(19) NOT NULL,
42     BranchBranchId number(19) NOT NULL,
43     PRIMARY KEY (HistoryID));
44
45   CREATE TABLE Role (
46     RoleID number(19) GENERATED AS IDENTITY,
47     BaseSalary number(10) NOT NULL CHECK(BaseSalary>0),
48     Name varchar2(255) NOT NULL,
49     Description varchar2(255) NOT NULL,
50     PRIMARY KEY (RoleID));
51
52   CREATE TABLE Loan (
53     LoanID number(19) GENERATED AS IDENTITY,
54     Amount number(19, 2) NOT NULL CHECK(Amount>0),
55     InterestRate number(3, 5) NOT NULL,
56     DateOfCompletion date NOT NULL,
57     DateOfCreation date NOT NULL,
58     BranchBranchId number(19) NOT NULL,
59     PRIMARY KEY (LoanID));
60
61   CREATE TABLE Payment (
62     PaymentID number(19) GENERATED AS IDENTITY,
63     PaymentDate date NOT NULL,
64     Amount number(19, 2) NOT NULL CHECK(Amount>0),
65     LoanLoanID number(19) NOT NULL,
66     CurrentAccountAccountID number(19) NOT NULL,
67     PRIMARY KEY (PaymentID));
68
```

```
69  CREATE TABLE SavingsAccount (
70    AccountAccountID number(19) NOT NULL,
71    InterestRate number(3, 5) NOT NULL CHECK(InterestRate>0),
72    TypeOfInterest varchar2(255) NOT NULL,
73    DurationYears number(10) NOT NULL CHECK(DurationYears>0),
74    PRIMARY KEY (AccountAccountID));
75
76  CREATE TABLE Deposit (
77    DepositID number(19) GENERATED AS IDENTITY,
78    DepositDate date NOT NULL,
79    Amount number(19, 2) NOT NULL CHECK(Amount>0),
80    AccountAccountID number(19) NOT NULL,
81    PRIMARY KEY (DepositID));
82
83  CREATE TABLE Tranfer (
84    TransferID number(19) GENERATED AS IDENTITY,
85    TransferDate date NOT NULL,
86    Amount number(19, 2) NOT NULL CHECK(Amount>0),
87    AccountAccountID number(19) NOT NULL,
88    AccountAccountID2 number(19) NOT NULL,
89    PRIMARY KEY (TransferID));
90
91  CREATE TABLE Withdraw (
92    WithdrawID number(19) GENERATED AS IDENTITY,
93    WithdrawDate date NOT NULL,
94    Amount number(19, 2) NOT NULL CHECK(Amount>0),
95    CurrentAccountAccountID number(19) NOT NULL,
96    PRIMARY KEY (WithdrawID));
97
98  CREATE TABLE CurrentAccount (
99    AccountAccountID number(19) NOT NULL,
100   Name varchar2(255) NOT NULL,
101   MaximumWithdraw number(10),
102   PRIMARY KEY (AccountAccountID));
103
104 ALTER TABLE Costumer ADD CONSTRAINT FKCostumer923053 FOREIGN KEY (PersonGovID) REFERENCES Person (
        ↪ GovID);
105
106 ALTER TABLE Employee ADD CONSTRAINT FKEmployee249023 FOREIGN KEY (PersonGovID) REFERENCES Person (
        ↪ GovID);
107
108 ALTER TABLE Account ADD CONSTRAINT FKAccount895601 FOREIGN KEY (CostumerCostumerID) REFERENCES
        ↪ Costumer (CostumerID);
109
110 ALTER TABLE RoleHistory ADD CONSTRAINT FKRoleHistor647811 FOREIGN KEY (RoleRoleID) REFERENCES Role (
        ↪ RoleID);
111
112 ALTER TABLE RoleHistory ADD CONSTRAINT FKRoleHistor516821 FOREIGN KEY (EmployeeEmployeeID) REFERENCES
        ↪ Employee (EmployeeID);
113
114 ALTER TABLE RoleHistory ADD CONSTRAINT FKRoleHistor171832 FOREIGN KEY (BranchBranchId) REFERENCES
        ↪ Branch (BranchId);
115
116 ALTER TABLE Loan ADD CONSTRAINT FKLoan357293 FOREIGN KEY (BranchBranchId) REFERENCES Branch (BranchId)
        ↪ ;
117
118 ALTER TABLE SavingsAccount ADD CONSTRAINT FKSavingsAcc25288 FOREIGN KEY (AccountAccountID) REFERENCES
        ↪ Account (AccountID);
119
120 ALTER TABLE Payment ADD CONSTRAINT FKPayment955503 FOREIGN KEY (LoanLoanID) REFERENCES Loan (LoanID);
121
```

```
122 ALTER TABLE Deposit ADD CONSTRAINT FKDeposit626030 FOREIGN KEY (AccountAccountID) REFERENCES Account (
        ↪ AccountID);
123
124 ALTER TABLE Tranfer ADD CONSTRAINT FKTranfer816388 FOREIGN KEY (AccountAccountID) REFERENCES Account (
        ↪ AccountID);
125
126 ALTER TABLE Tranfer ADD CONSTRAINT FKTranfer299653 FOREIGN KEY (AccountAccountID2) REFERENCES Account
        ↪ (AccountID);
127
128 ALTER TABLE Payment ADD CONSTRAINT FKPayment25568 FOREIGN KEY (CurrentAccountAccountID) REFERENCES
        ↪ CurrentAccount (AccountAccountID);
129
130 ALTER TABLE Withdraw ADD CONSTRAINT FKWithdraw546165 FOREIGN KEY (CurrentAccountAccountID) REFERENCES
        ↪ CurrentAccount (AccountAccountID);
131
132 ALTER TABLE CurrentAccount ADD CONSTRAINT FKCurrentAcc16041 FOREIGN KEY (AccountAccountID) REFERENCES
        ↪ Account (AccountID);
```

Transactions

1:Changing Query

Description

There is 5 CurrentAccount in the system. This transaction doubles the amount of the account that has the biggest value in the database.

Input

Output

SQL

```
1  BEGIN TRANSACTION
2  UPDATE Account
3  SET amount = amount*2
4  WHERE AccountID = (
5      SELECT AccountID
6      FROM Costumer c
7      INNER JOIN Person p
8          ON p.GovID = c.PersonGovID
9      INNER JOIN Account
10         ON CostumerID = CostumerCostumerID
11     INNER JOIN CurrentAccount
12         ON AccountAccountID = AccountID
13     WHERE amount >= ALL(
14         SELECT MAX(amount)
15         FROM Account
16     )
17 );
18 COMMIT
```

2:Changing Query

Description

This transaction preforms a withdraw of 100 units on the CurrentAccount with the AccountAccountID 1. To do this we must check if the amount we want to withdraw is smaller than theCurrentAccount MaximumWithdraw. After that we update the amount and add an entry to the Withdraw ledger.

Input

Output

SQL

```
BEGIN TRANSACTION
UPDATE Account
SET amount =
    CASE
        WHEN 100<(
            SELECT MaximumWithdraw
            FROM CurrentAccount
            WHERE AccountAccountID=1)
          THEN amount - 100
        ELSE amount
    END
WHERE AccountID=(
    SELECT AccountAccountID
    FROM CurrentAccount INNER JOIN Account
        ON AccountID=AccountAccountID
    WHERE AccountAccountID=1);

UPDATE Account
SET EndDate =
    CASE
        WHEN amount=0 THEN CURRENT_DATE
        ELSE null
    END
WHERE AccountID = (
    SELECT AccountAccountID
    FROM CurrentAccount INNER JOIN Account
        ON AccountID=AccountAccountID
    WHERE AccountAccountID=1);

INSERT INTO Withdraw(WithdrawDate, Amount, CurrentAccountAccountID) VALUES(CURRENT_DATE, 100, 1);
COMMIT
```