

AUTENTICAÇÃO E SEGURANÇA NOS SGBD E NAS BD - EXEMPLO

Objetivos

- Implementar políticas de segurança em Sistemas de Gestão de Bases de Dados Relacionais
- Definir políticas de segurança e acesso aos dados

Descrição do problema

Pretende-se desenvolver uma aplicação simplificada para exemplificar o funcionamento de um banco.

O modelo físico da Base de Dados é fornecido em anexo.

Pretende-se criar um modelo de segurança para a base de dados que garanta o acesso a quatro tipos de utilizadores:

- Cliente (acesso de *home-banking*);
- Caixa;
- Gerente;
- Administrador do sistema.

O cliente deverá ter acesso a todas as suas contas a partir do seu *home-banking*. Só deverão ser permitidas operações de consulta.

O caixa, para além de poder consultar toda a informação, pode também efetuar depósitos e levantamentos.

O gerente pode criar novos clientes e abrir as respetivas contas.

O administrador terá permissões para gerir toda a informação do banco.

Código para geração da base de dados

```
USE master
GO
```

```
CREATE DATABASE bd_banco
GO
```

```
USE bd_banco
GO
```

```
CREATE TABLE CodigoPostal(
    cod_postal    CHAR(8)        NOT NULL,
    localidade    VARCHAR(30)    NOT NULL,
    PRIMARY KEY(cod_postal)
)
```

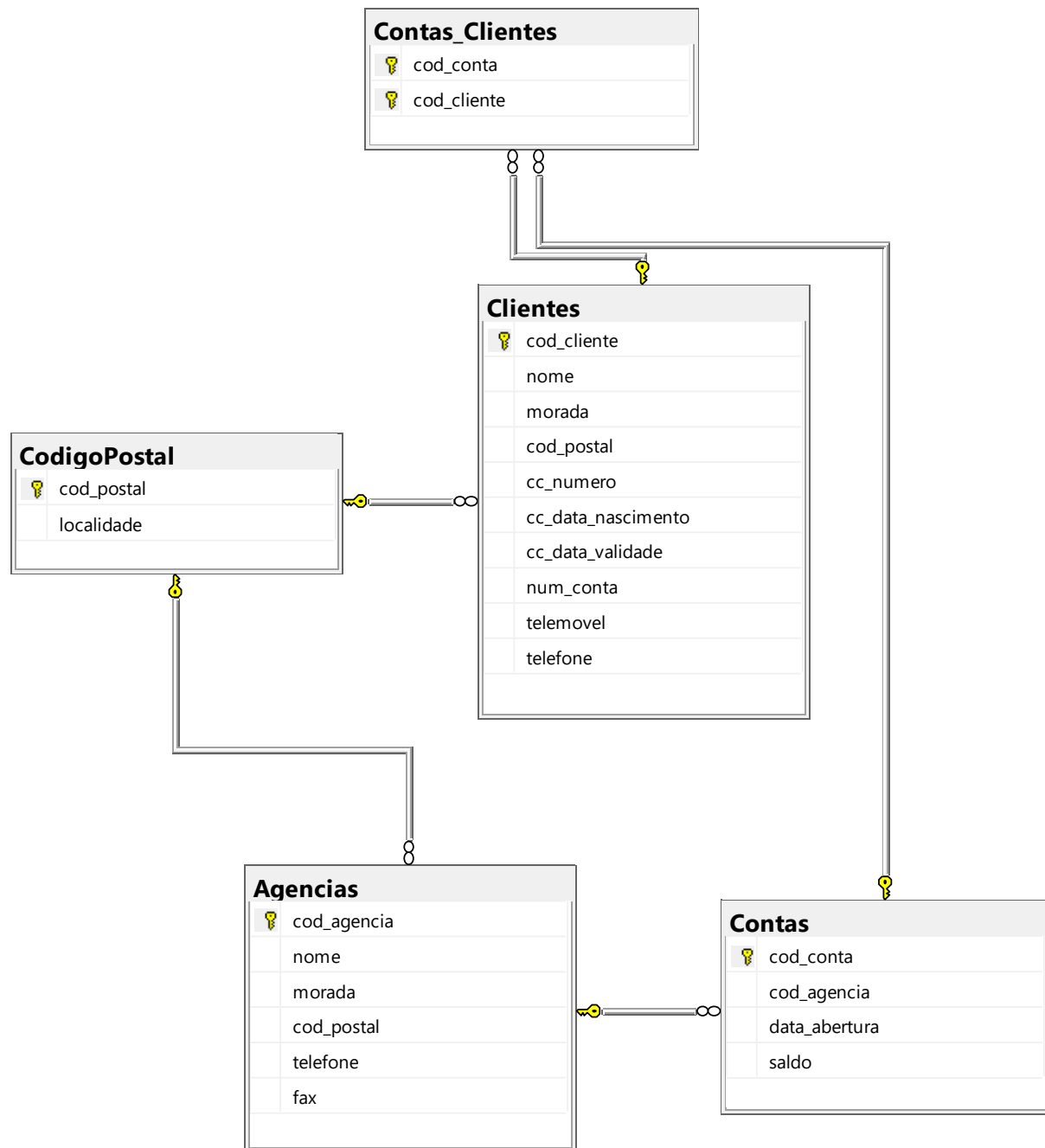
```
CREATE TABLE Agencias(
    cod_agencia   INTEGER        NOT NULL,
    nome          VARCHAR(50)    NOT NULL,
    morada        VARCHAR(50)    NOT NULL,
    cod_postal    CHAR(8)        NOT NULL,
    telefone      CHAR(9)        NOT NULL,
    fax           CHAR(9)        NOT NULL,
    PRIMARY KEY(cod_agencia),
    FOREIGN KEY(cod_postal) REFERENCES CodigoPostal(cod_postal)
)
```

```
CREATE TABLE Clientes(
    cod_cliente   INTEGER        NOT NULL,
    nome          VARCHAR(50)    NOT NULL,
    morada        VARCHAR(50)    NOT NULL,
    cod_postal    CHAR(8)        NOT NULL,
    -- cc -> cartao cidadão
    cc_numero     CHAR(12)       NOT NULL,
    cc_data_nascimento DATE      NOT NULL,
    cc_data_validade DATE      NOT NULL,
    num_conta     CHAR(13)       NOT NULL,
    telemovel     CHAR(9)        NOT NULL,
    telefone      CHAR(9),
    PRIMARY KEY(cod_cliente),
    FOREIGN KEY(cod_postal) REFERENCES CodigoPostal(cod_postal)
)
```

```
CREATE TABLE Contas(
    cod_conta     INTEGER        NOT NULL,
    cod_agencia   INTEGER        NOT NULL,
    data_abertura DATE          NOT NULL,
    saldo        MONEY          NOT NULL,
    PRIMARY KEY(cod_conta),
    FOREIGN KEY(cod_agencia) REFERENCES Agencias(cod_agencia)
)
```

```
CREATE TABLE Contas_Clientes(
    cod_conta     INTEGER        NOT NULL,
    cod_cliente   INTEGER        NOT NULL,
    PRIMARY KEY(cod_conta, cod_cliente),
    FOREIGN KEY(cod_conta) REFERENCES Contas(cod_conta),
    FOREIGN KEY(cod_cliente) REFERENCES Clientes(cod_cliente)
)
```

Diagrama da base de dados



Solução

```
-- Criar Logins
USE master
GO

CREATE LOGIN ClienteL WITH PASSWORD = '123456'
CREATE LOGIN CaixaL WITH PASSWORD = '123456'
CREATE LOGIN GerenteL WITH PASSWORD = '123456'
CREATE LOGIN AdministradorL WITH PASSWORD = '123456'

--Criar Users
USE bd_banco
GO

CREATE USER ClienteU FOR LOGIN ClienteL
CREATE USER CaixaU FOR LOGIN CaixaL
CREATE USER GerenteU FOR LOGIN GerenteL
CREATE USER AdministradorU FOR LOGIN AdministradorL

--Definir permissões

--Clientes
GRANT SELECT TO ClienteU

--Caixas
GRANT SELECT TO CaixaU
GRANT UPDATE ON Contas(saldo) TO CaixaU

--Gerentes
GRANT INSERT ON Clientes TO GerenteU
GRANT INSERT ON Contas TO GerenteU
GRANT INSERT ON Contas_Clientes TO GerenteU
GRANT SELECT ON CodigoPostal TO GerenteU
GRANT SELECT ON Agencias TO GerenteU

--Administradores
ALTER ROLE db_owner ADD MEMBER AdministradorU
--Ou
GRANT SELECT, INSERT, UPDATE, DELETE TO AdministradorU
```