```
CREATE TABLE branches (
        address varchar(50)
);
CREATE TABLE employees (
        namez char(20),
        position1 char(20),
        address varchar(50),
        SSN varchar(11),
        salary int(100),
        branchname varchar(20),

        PRIMARY KEY (namez, SSN),
        FOREIGN KEY (branchname) references branches,
        CHECK (branchname is not NULL AND salary > 0)
);

CREATE TABLE customers (
        customerID INT(20),
        name1 char(20),
        addresses varchar(50),
        homebranch varchar(20),

        PRIMARY KEY (name1, addresses),
        FOREIGN KEY (homebranch) references branches,
        FOREIGN KEY (customerID) references accounts,
        CHECK (homebranch is not NULL)

);

CREATE TABLE accounts (
        customerID INT(20),
        transactionID INT(20),
        IsChecking BIT,
        IsSaving BIT,
        overDraftFee INT(10),
        monthlyFee INT(10),
        accountNumber INT(20),
        currentBalance INT(20),
        PRIMARY KEY (accountNumber),
        CHECK (currentBalance > 0)

);

CREATE TABLE transactions(
```

```
        transactionID INT(20),
        transactionType char(20),
        amount numeric(20),
        description varchar(100),
        PRIMARY KEY (amount, description),
        FOREIGN KEY (transactionID) references accounts
);

CREATE TABLE loans(
        customerID INT(20),
        loanAmount INT(20),
        runtime varchar(20),
        interestSchedule numeric(20),
        PRIMARY KEY (loanAmount, runtime, interestSchedule),
        FOREIGN KEY (customerID) references accounts
);

CREATE ROLE customer;
GRANT READ ON account to customer;

CREATE ROLE tellers;
GRANT READ ON account TO tellers;
GRANT UPDATE ON transactions TO Tellers;

CREATE ROLE manager;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO manager;

CREATE ROLE loanManager;
GRANT ALL PRIVILEGES ON customers TO loanManager;
GRANT ALL PRIVILEGES ON loans TO loanManager;
```