

## قسمت اول و دوم:

چک کردن عملکرد صحیح متد های طراحی شده (عملیات CRUD):

```

> main.go > ⌂ main
1 func main() {
2     customerLogin := &model.CustomerLogin{
3         Username: "davidmills",
4         Password: "password2627",
5         CustomerID: 0, //will be set by database
6     }
7
8     customerEmail := &model.CustomerEmail{
9         Email: "davidmillss8@gmail.com",
10    }
11
12    customerDetails := &model.CustomerDetails{
13        FirstName: "david",
14        LastName: "mills",
15        BirthDate: "1985-11-02",
16        PhoneNumber: "214555123",
17        Address: "12 main st",
18        CustomerType: "Natural", //haghghi
19        BankID: 1,
20    }
21    db.RegisterCustomer(customerLogin, customerEmail, customerDetails)
22
23    customer, err := db.LoginCustomer("davidmills", "password2627")
24    if err != nil {
25        log.Fatal(err)
26    }
27    fmt.Printf("logged in customer: %+v\n", customer)
28
29    accountNumber := &model.AccountNumber{
30        AccountNumber: "1921072918",
31        AccountID: 0, //will be set by database
32    }
33
34    accountDetails := &model.AccountDetails{
35        AccountType: "Deposit",
36        AccountPassword: "1111",
37        Balance: 5000,
38        AccountStatus: "Active",
39        OpenDate: "2025-01-01",
40        CustomerID: 1,
41    }
42    db.CreateAccount(accountNumber, accountDetails)

```

```

func main() {
    customerLogin2 := &model.CustomerLogin{
        Username: "sallysmith",
        Password: "salsal",
        CustomerID: 0, //will be set by database
    }

    customerEmail2 := &model.CustomerEmail{
        Email: "sallysmith345@gmail.com",
    }

    customerDetails2 := &model.CustomerDetails{
        FirstName: "sally",
        LastName: "smith",
        BirthDate: "1982-10-13",
        PhoneNumber: "3425534657",
        Address: "18 main st",
        CustomerType: "Legal", //hoghoghi
        BankID: 1,
    }
    db.RegisterCustomer(customerLogin2, customerEmail2, customerDetails2)

    accountNumber2 := &model.AccountNumber{
        AccountNumber: "3424235456",
        AccountID: 0, //will be set by database
    }

    accountDetails2 := &model.AccountDetails{
        AccountType: "Deposit",
        AccountPassword: "1234",
        Balance: 1000,
        AccountStatus: "Closed",
        OpenDate: "2014-03-01",
        CustomerID: 2,
    }

    db.CreateAccount(accountNumber2, accountDetails2)

    db.GetAccount("1921072918", "1111")
    db.GetAccount("3424235456", "1234")
}

```

```

postgres | 2025-01-03 09:59:41.693 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
postgres | 2025-01-03 09:59:41.693 UTC [1] LOG:  listening on IPv6 address "::", port 5432
postgres | 2025-01-03 09:59:41.695 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres | 2025-01-03 09:59:41.697 UTC [65] LOG:  database system was shut down at 2025-01-03 09:59:41 UTC
postgres | 2025-01-03 09:59:41.703 UTC [1] LOG:  database system is ready to accept connections
postgres | 2025-01-03 09:59:45.801 UTC [79] FATAL:  database "neda.z" does not exist
app | connected to database
app | Successfully created bank
app | successfully created branch
app | Successfully registered employee
app | Successfully registered customer
app | logged in customer: &{CustomerID:1 FirstName:david LastName:mills BirthDate:1985-11-02T00:00:00Z PhoneNumber:214
555123 Address:12 main st CustomerType:Natural BankID:1}
app | successfully created account
app | getting the customer 1 account:
app | Starting server on :8080

```

```

psql (13.17 (Debian 13.17-1.pgdg120+1))
Type "help" for help.

banking_system=# select * from customer_details;
 customer_id | first_name | last_name | birth_date | phone_number | address      | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+
      1 | david      | mills      | 1985-11-02 | 214555123   | 12 main st | Natural      |      1
(1 row)

banking_system=# select * from account_details;
 account_id | customer_id | account_type | account_password | balance | account_status | open_date    | close_date
-----+-----+-----+-----+-----+-----+-----+-----+
      1 |           1 | Deposit      | 1111            | 5000.00 | Active       | 2025-01-01 |
(1 row)

```

```

banking_system=# select * from customer_details;
 customer_id | first_name | last_name | birth_date | phone_number | address      | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+
      1 | david      | mills      | 1985-11-02 | 214555123   | 12 main st | Natural      |      1
(1 row)

banking_system=# UPDATE customer_details SET address = '456 Elm St' WHERE customer_id = 1;
UPDATE 1
banking_system=# select * from customer_details;
 customer_id | first_name | last_name | birth_date | phone_number | address      | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+
      1 | david      | mills      | 1985-11-02 | 214555123   | 456 Elm St | Natural      |      1
(1 row)

```

```

banking_system=# delete from customer_details where customer_id = 1;
DELETE 1
banking_system=# select * from customer_details;
 customer_id | first_name | last_name | birth_date | phone_number | address      | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+
(0 rows)

```

```

→ DBproject git:(master) ✘ docker exec -it postgres psql -U neda.z -d banking_system

psql (13.17 (Debian 13.17-1.pgdg120+1))
Type "help" for help.

banking_system=# select * from customer_details;
 customer_id | first_name | last_name | birth_date | phone_number | address      | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+
      1 | david      | mills      | 1985-11-02 | 214555123   | 12 main st | Natural      |      1
      2 | sally      | smith      | 1982-10-13 | 3425534657  | 18 main st | Legal        |      1
(2 rows)

banking_system=# select * from account_details;
 account_id | customer_id | account_type | account_password | balance | account_status | open_date    | close_date
-----+-----+-----+-----+-----+-----+-----+-----+
      1 |           1 | Deposit      | 1111            | 5000.00 | Active       | 2025-01-01 |
      2 |           2 | Deposit      | 1234            | 1000.00 | Closed       | 2014-03-01 |
(2 rows)

banking_system=# select * from account_details where account_status = 'Active';
 account_id | customer_id | account_type | account_password | balance | account_status | open_date    | close_date
-----+-----+-----+-----+-----+-----+-----+-----+
      1 |           1 | Deposit      | 1111            | 5000.00 | Active       | 2025-01-01 |
(1 row)

```

همچنین برای افزایش security پسورد ها را به صورت هش شده در جداول ذخیره میکنم:

| account_id | customer_id | account_type | account_password   | balance | account_status |
|------------|-------------|--------------|--|---------|----------------|
|            |             |              |  |         |                |
| 1          | 1           | Deposit      | \$2a\$10\$1/QJUCZ1M.SZ.c30vJ5dr.rjQ379w6PwIIiCg.BnwRgq4ZkDctMIRO | 5000.00 | Active         |
| 2025-01-01 | 2           | Deposit      | \$2a\$10\$3soC8xq7sX3B6JLYGAAHHuEgPzBdQ5w.LsYQFpdsJ/BQsMcXr7Kh2  | 1000.00 | Closed         |
| 2014-03-01 |             |              |  |         |                |

### قسمت سوم:

#### ۱. اضافه کردن یک customer جدید:

```

banking_system=# select * from customer_details;
+-----+-----+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | birth_date | phone_number | address | customer_type | bank_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | david | mills | 1985-11-02 | 214555123 | 12 main st | Natural | 1 |
| 2 | sally | smith | 1982-10-13 | 3425534657 | 18 main st | Legal | 1 |
+-----+-----+-----+-----+-----+-----+-----+
(2 rows)

banking_system=# select * from customer_login;
+-----+-----+-----+
| username | password | customer_id |
+-----+-----+-----+
| davidmills | $2a$10$TGl1kHa7HEV.VQC46P4wuBx59U7jj52psi5y40y/560Xmw9js722 | 1 |
| sallysmith | $2a$10$3UEFz1mu/VjV3pVYJRV0X.4i/D0zoRilKmtVmZlWj66cbcVjf4XiA | 2 |
+-----+-----+-----+
(2 rows)

banking system=# INSERT INTO customer_login (username, password, customer_id)
VALUES ('markrobertsss', 'password', DEFAULT);
INSERT 0 1
banking_system=# select * from customer_login;
+-----+-----+-----+-----+
| username | password | customer_id |
+-----+-----+-----+
| davidmills | $2a$10$TGl1kHa7HEV.VQC46P4wuBx59U7jj52psi5y40y/560Xmw9js722 | 1 |
| sallysmith | $2a$10$3UEFz1mu/VjV3pVYJRV0X.4i/D0zoRilKmtVmZlWj66cbcVjf4XiA | 2 |
| markrobertsss | password | 3 |
+-----+-----+-----+
(3 rows)

banking system=# INSERT INTO customer_details (customer_id, first_name, last_name, birth_date, phone_number, address, customer_type, bank_id)
VALUES (currval('customer_login_customer_id_seq'), 'mark', 'roberts', '1956-01-01', '1234567890', '56 Main St', 'Natural', 1 );
INSERT 0 1
banking_system=# select * from customer_details;
+-----+-----+-----+-----+-----+-----+-----+
| customer_id | first_name | last_name | birth_date | phone_number | address | customer_type | bank_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | david | mills | 1985-11-02 | 214555123 | 12 main st | Natural | 1 |
| 2 | sally | smith | 1982-10-13 | 3425534657 | 18 main st | Legal | 1 |
| 3 | mark | roberts | 1956-01-01 | 1234567890 | 56 Main St | Natural | 1 |
+-----+-----+-----+-----+-----+-----+-----+
(3 rows)

```

## ۲. افزودن حساب جدید برای کاستومر جدید

```
banking_system=# select * from account_numbers;
account_number | account_id
-----+-----
1921072918    |      1
3424235456    |      2
(2 rows)

banking_system=# INSERT INTO account_numbers (account_number, account_id)
VALUES ('6534123456', DEFAULT);
INSERT 0 1
banking_system=# select * from account_numbers;
account_number | account_id
-----+-----
1921072918    |      1
3424235456    |      2
6534123456    |      3
(3 rows)

banking_system=# select * from account_details;
banking_system=# INSERT INTO account_details (account_id, customer_id, account_type, account_password, balance, open_date)
VALUES (curval('account_numbers_account_id_seq'), 3, 'Savings', '2456', 4500, CURRENT_DATE);
INSERT 0 1
```

| account_id | customer_id | account_type | account_password  | balance | account_ |
|------------|-------------|--------------|---|---------|----------|
| status     | open_date   | close_date   |   |         |          |
| 1          | 1           | Deposit      | \$2a\$10\$CbCWifdnknfYLAVqyZnM/u0nsAyni.2xTgMZf2ICcS4Icc5Aonek2 | 5000.00 | Active   |
| 2          | 2           | Deposit      | \$2a\$10\$GbDnve2oWlMyzvXA4fmazekNfNnA20xyUNDkUgMkHcQ5UFRBgBFv6 | 1000.00 | Closed   |
| 3          | 3           | Savings      | 2456  | 4500.00 | Active   |
|            |             |              |   |         |          |

## ۳. انتخاب و نمایش همه تراکنش های یک حساب خاص

اول یکسری تراکنش ایجاد میکنیم:

```
banking_system=# INSERT INTO transaction (source_account_id, destination_account_id, amount, transaction_type, transaction_date)
SELECT ad.account_id, NULL, 1000.00, 'Deposit', CURRENT_TIMESTAMP
FROM account_details ad
WHERE ad.customer_id = 3
LIMIT 1;
INSERT 0 1
banking_system=# select * from transaction;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----+
1 | 1 | 3 | 1000.00 | Deposit | 2025-01-03 15:57:37
(1 row)

banking_system=# UPDATE account_details
SET balance = balance + 1000.00
WHERE customer_id = 3;
UPDATE 1
banking_system=# INSERT INTO transaction (source_account_id, destination_account_id, amount, transaction_type, transaction_date)
SELECT ad.account_id, NULL, 500.00, 'Withdrawal', CURRENT_TIMESTAMP
FROM account_details ad
WHERE ad.customer_id = 3
LIMIT 1;
INSERT 0 1
banking_system=# select * from transaction;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----+
1 | 1 | 3 | 1000.00 | Deposit | 2025-01-03 15:57:37
2 | 2 | 3 | 500.00 | Withdrawal | 2025-01-03 15:59:24
(2 rows)

banking_system=# UPDATE account_details
SET balance = balance - 500.00
WHERE customer_id = 3;
UPDATE 1
```

```

banking_system=# INSERT INTO transaction (source_account_id, destination_account_id, amount, transaction_type, transaction_date)
SELECT ad.account_id, 5, 300.00, 'Transfer', CURRENT_TIMESTAMP
FROM account_details ad
WHERE ad.customer_id = 3
LIMIT 1;
ERROR: insert or update on table "transaction" violates foreign key constraint "transaction_destination_account_id_foreign"
DETAIL: Key (destination_account_id)=(5) is not present in table "account_numbers".
banking_system=# INSERT INTO transaction (source_account_id, destination_account_id, amount, transaction_type, transaction_date)

SELECT ad.account_id, 1, 300.00, 'Transfer', CURRENT_TIMESTAMP
FROM account_details ad
WHERE ad.customer_id = 3
LIMIT 1;
INSERT 0 1
banking_system=# select * from transaction;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----+
1 | 3 | 3 | 1000.00 | Deposit | 2025-01-03 15:57:37
2 | 3 | 3 | 500.00 | Withdrawal | 2025-01-03 15:59:24
4 | 3 | 3 | 300.00 | Transfer | 2025-01-03 16:01:15
(3 rows)

```

```

banking_system=# UPDATE account_details
SET balance = balance - 300.00
WHERE customer_id = 3;
UPDATE 1
banking_system=# UPDATE account_details
SET balance = balance + 300.00
WHERE account_id = 1;
UPDATE 1
banking_system=# select * from account_details;

```

| account_id | customer_id | account_type | account_password  | balance | account_status | open_date  |
|------------|-------------|--------------|---|---------|----------------|------------|
| 2          | 2           | Deposit      | \$2a\$10\$GbDnve2oWlMyzvXA4fmazekNfNnA20xyUNDkUgMkHcQ5UFRBgBFv6 | 1000.00 | Closed         | 2014-03-01 |
| 3          | 3           | Savings      | 2456  | 4700.00 | Active         | 2025-01-03 |
| 1          | 1           | Deposit      | \$2a\$10\$CbCWifdnknfYLAVqyZnM/u0nsAyni.2xTgMZF2ICcs4Icc5Aonek2 | 5300.00 | Active         | 2025-01-01 |

(3 rows)

حال همه تراکنش های کاستومر ۱ ( حساب ۱ ) و کاستومر ۳ ( حساب ۳ ) رو میبینیم:

```

banking_system=# SELECT t.*
FROM transaction t
WHERE t.source_account_id = 3
    OR t.destination_account_id = 3
ORDER BY t.transaction_date DESC;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----+
        4 |            3 |            3 | 300.00 | Transfer | 2025-01-03 16:01:15
        2 |            3 |            3 | 500.00 | Withdrawal | 2025-01-03 15:59:24
        1 |            3 |            3 | 1000.00 | Deposit | 2025-01-03 15:57:37
(3 rows)

banking_system=# SELECT t.*
FROM transaction t
WHERE t.source_account_id = 1
    OR t.destination_account_id = 1
ORDER BY t.transaction_date DESC;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----+
        4 |            3 |            3 | 300.00 | Transfer | 2025-01-03 16:01:15
(1 row)

```

#### ۴. همه وام های فعال در loan:

اول پکسری وام ایجاد میکنیم:

```
banking_system=# INSERT INTO loan (customer_id, loan_type, amount, interest_rate, duration, start_date, end_date, loan_status)
VALUES
(1, 'Home', 250000.00, 4.5, 360, '2024-01-01', '2054-01-01', 'Active'),
(1, 'Car', 35000.00, 6.0, 60, '2024-01-01', '2029-01-01', 'Active'),
(2, 'Personal', 10000.00, 8.5, 24, '2024-01-01', '2026-01-01', 'Active'),
(3, 'Business', 100000.00, 7.0, 120, '2024-01-01', '2034-01-01', 'Active'),
(3, 'Education', 50000.00, 5.5, 84, '2024-01-01', '2031-01-01', 'Paidoff');
INSERT 0 5
banking_system=# select * from loan;
+-----+-----+-----+-----+-----+-----+-----+-----+
| loan_id | customer_id | loan_type | amount | interest_rate | duration | start_date | end_date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 11 | 1 | Home | 250000.00 | 4.50 | 360 | 2024-01-01 | 2054-01-01 | Active
| 12 | 1 | Car | 35000.00 | 6.00 | 60 | 2024-01-01 | 2029-01-01 | Active
| 13 | 2 | Personal | 10000.00 | 8.50 | 24 | 2024-01-01 | 2026-01-01 | Active
| 14 | 3 | Business | 100000.00 | 7.00 | 120 | 2024-01-01 | 2034-01-01 | Active
| 15 | 3 | Education | 50000.00 | 5.50 | 84 | 2024-01-01 | 2031-01-01 | Paidoff
+-----+-----+-----+-----+-----+-----+-----+-----+
(5 rows)
```

```
banking_system=# SELECT *
FROM loan
WHERE loan_status = 'Active';
+-----+-----+-----+-----+-----+-----+-----+-----+
| loan_id | customer_id | loan_type | amount | interest_rate | duration | start_date | end_date | loan_status
+-----+-----+-----+-----+-----+-----+-----+-----+
| 11 | 1 | Home | 250000.00 | 4.50 | 360 | 2024-01-01 | 2054-01-01 | Active
| 12 | 1 | Car | 35000.00 | 6.00 | 60 | 2024-01-01 | 2029-01-01 | Active
| 13 | 2 | Personal | 10000.00 | 8.50 | 24 | 2024-01-01 | 2026-01-01 | Active
| 14 | 3 | Business | 100000.00 | 7.00 | 120 | 2024-01-01 | 2034-01-01 | Active
+-----+-----+-----+-----+-----+-----+-----+-----+
(4 rows)
```

```
banking_system=# INSERT INTO loanPayment (loan_id, payment_amount, due_date, payment_date, payment_status)
SELECT
    loan_id,
    amount / duration as payment_amount,
    start_date + interval '1 month' as due_date,
    start_date + interval '1 month' as payment_date,
    'Paid' as payment_status
FROM loan;
INSERT 0 5
banking_system=# select * from loanPayment;
+-----+-----+-----+-----+-----+
| payment_id | loan_id | payment_amount | due_date | payment_date | payment_status
+-----+-----+-----+-----+-----+
| 1 | 11 | 694.44 | 2024-02-01 | 2024-02-01 | Paid
| 2 | 12 | 583.33 | 2024-02-01 | 2024-02-01 | Paid
| 3 | 13 | 416.67 | 2024-02-01 | 2024-02-01 | Paid
| 4 | 14 | 833.33 | 2024-02-01 | 2024-02-01 | Paid
| 5 | 15 | 595.24 | 2024-02-01 | 2024-02-01 | Paid
+-----+-----+-----+-----+-----+
(5 rows)
```

#### ۵. همه حساب هایی که مقدارشان از 5000 بیشتر است:

```
banking_system=# SELECT ad.*, an.account_number
FROM account_details ad
JOIN account_numbers an ON ad.account_id = an.account_id
WHERE ad.balance > 5000;
```

| account_id | customer_id | account_type | close_date | account_number | account_password  | balance | account_status | open_date  |
|------------|-------------|--------------|------------|----------------|---|---------|----------------|------------|
| 1          | 1           | Deposit      |            | 1921072918     | \$2a\$10\$CbCWifdnknfYLAVqyZnM/u0nsAyni.2xTgMZF2ICcS4Icc5Aonek2 | 5300.00 | Active         | 2025-01-01 |

(1 row)

.7

```
banking_system=# SELECT ad.*, an.account_number
  FROM account_details ad
  JOIN account_numbers an ON ad.account_id = an.account_id
 WHERE ad.balance > 5000;
banking_system=# SELECT
    cd.first_name,
    cd.last_name,
    ad.account_type,
    SUM(ad.balance) as total_balance
  FROM customer_details cd
  JOIN account_details ad ON cd.customer_id = ad.customer_id
  JOIN account_numbers an ON ad.account_id = an.account_id
 GROUP BY cd.customer_id, cd.first_name, cd.last_name, ad.account_type;
   first_name | last_name | account_type | total_balance
-----+-----+-----+-----
  mark      | roberts    | Savings        |      4700.00
 sally      | smith       | Deposit        |      1000.00
 david      | mills       | Deposit        |      5300.00
(3 rows)
```

.V

```
banking_system=# SELECT
    ed.first_name,
    ed.last_name,
    SUM(l.amount) as total_loan_amount
  FROM employee_details ed
  JOIN customer_details cd ON CONCAT(ed.first_name, ed.last_name) = CONCAT(cd.first_name, cd.last_name)
  JOIN loan l ON cd.customer_id = l.customer_id
 WHERE l.loan_status = 'Active'
 GROUP BY ed.employee_id, ed.first_name, ed.last_name;
   first_name | last_name | total_loan_amount
-----+-----+-----
(0 rows)
```

.A

```
banking_system=# SELECT
    cd.first_name,
    cd.last_name,
    COUNT(ad.account_id) as account_count
  FROM customer_details cd
  JOIN account_details ad ON cd.customer_id = ad.customer_id
 GROUP BY cd.customer_id, cd.first_name, cd.last_name
 HAVING COUNT(ad.account_id) > 1;
   first_name | last_name | account_count
-----+-----+-----
(0 rows)
```

بخش امتیازی:

```
banking_system=# SELECT
    cd.first_name,
    cd.last_name,
    COUNT(l.loan_id) as active_loan_count
FROM customer_details cd
JOIN loan l ON cd.customer_id = l.customer_id
WHERE l.loan_status = 'Active'
GROUP BY cd.customer_id, cd.first_name, cd.last_name
ORDER BY active_loan_count DESC
LIMIT 5;
first_name | last_name | active_loan_count
-----+-----+-----
david      | mills      |          2
sally      | smith      |          1
mark       | roberts    |          1
(3 rows)
```

```
banking_system=# SELECT
    l.*,
    COUNT(lp.payment_id) as paid_installments
FROM loan l
LEFT JOIN loanPayment lp ON l.loan_id = lp.loan_id
WHERE lp.payment_status = 'Paid'
GROUP BY l.loan_id
ORDER BY paid_installments ASC
LIMIT 5;
```

| loan_id | customer_id | loan_type | amount    | interest_rate | duration | start_date | end_date   | loan_status | paid_installments |
|---------|-------------|-----------|-----------|---------------|----------|------------|------------|-------------|-------------------|
| 11      | 1           | Home      | 250000.00 | 4.50          | 360      | 2024-01-01 | 2054-01-01 | Active      | 1                 |
| 12      | 1           | Car       | 35000.00  | 6.00          | 60       | 2024-01-01 | 2029-01-01 | Active      | 1                 |
| 13      | 2           | Personal  | 10000.00  | 8.50          | 24       | 2024-01-01 | 2026-01-01 | Active      | 1                 |
| 14      | 3           | Business  | 100000.00 | 7.00          | 120      | 2024-01-01 | 2034-01-01 | Active      | 1                 |
| 15      | 3           | Education | 50000.00  | 5.50          | 84       | 2024-01-01 | 2031-01-01 | Paidoff     | 1                 |

(5 rows)

```

banking_system=# SELECT
    cd.first_name,
    cd.last_name,
    l.loan_id,
    l.amount as loan_amount
FROM customer_details cd
JOIN loan l ON cd.customer_id = l.customer_id
JOIN loanPayment lp ON l.loan_id = lp.loan_id
WHERE lp.payment_status = 'Unpaid'
AND lp.due_date < CURRENT_DATE;
first_name | last_name | loan_id | loan_amount
-----+-----+-----+
(0 rows)

banking_system=# select * from loanPayment;
payment_id | loan_id | payment_amount | due_date | payment_date | payment_status
-----+-----+-----+-----+-----+-----+
1 | 11 | 694.44 | 2024-02-01 | 2024-02-01 | Paid
2 | 12 | 583.33 | 2024-02-01 | 2024-02-01 | Paid
3 | 13 | 416.67 | 2024-02-01 | 2024-02-01 | Paid
4 | 14 | 833.33 | 2024-02-01 | 2024-02-01 | Paid
5 | 15 | 595.24 | 2024-02-01 | 2024-02-01 | Paid
(5 rows)

```

```

banking_system=# SELECT
    cd.first_name,
    cd.last_name,
    SUM(ad.balance) as total_balance
FROM customer_details cd
JOIN account_details ad ON cd.customer_id = ad.customer_id
GROUP BY cd.customer_id, cd.first_name, cd.last_name
ORDER BY total_balance DESC
LIMIT 5;
first_name | last_name | total_balance
-----+-----+-----+
david | mills | 5300.00
mark | roberts | 4700.00
sally | smith | 1000.00
(3 rows)

```

قسمت چهارم:  
view های ایجاد شده:

```
banking_system=# CREATE VIEW customer_accounts AS
SELECT
    cd.first_name || ' ' || cd.last_name AS customer_name,
    cd.phone_number,
    an.account_number,
    ad.account_type,
    ad.balance
FROM customer_details cd
JOIN customer_login cl ON cd.customer_id = cl.customer_id
JOIN account_details ad ON cl.customer_id = ad.customer_id
JOIN account_numbers an ON ad.account_id = an.account_id;
CREATE VIEW
banking_system=# CREATE VIEW bank_transactions AS
SELECT
    b.name AS bank_name,
    t.transaction_id,
    src_an.account_number AS source_account_number,
    dst_an.account_number AS destination_account_number,
    t.amount AS transaction_amount,
    t.transaction_date
FROM transaction t
JOIN account_numbers src_an ON t.source_account_id = src_an.account_id
JOIN account_details src_ad ON src_an.account_id = src_ad.account_id
JOIN customer_details src_cd ON src_ad.customer_id = src_cd.customer_id
JOIN bank b ON src_cd.bank_id = b.bank_id
LEFT JOIN account_numbers dst_an ON t.destination_account_id = dst_an.account_id;
CREATE VIEW
banking_system=# CREATE VIEW bank_member AS
SELECT
    b.name AS bank_name,
    COALESCE(ed.first_name || ' ' || ed.last_name, cd.first_name || ' ' || cd.last_name) AS full_name,
    COALESCE(el.employee_id::TEXT, cl.customer_id::TEXT) AS member_id,
    CASE
        WHEN ed.employee_id IS NOT NULL THEN 'Employee'
        ELSE 'Customer'
    END AS role,
    COALESCE(ce.email, '') AS email,
    COALESCE(ed.position, '') AS position,
    COALESCE(cd.phone_number, '') AS phone_number
FROM bank b
LEFT JOIN employee_details ed ON ed.branch_id IN (SELECT branch_id FROM branch WHERE bank_id = b.bank_id)
LEFT JOIN employee_login el ON ed.employee_id = el.employee_id
LEFT JOIN customer_details cd ON cd.bank_id = b.bank_id
LEFT JOIN customer_login cl ON cd.customer_id = cl.customer_id
LEFT JOIN customer_email ce ON cl.customer_id = ce.customer_id;
CREATE VIEW
```

قسمت پنجم:

```
banking_system=# CREATE USER Blazkowicz WITH PASSWORD 'William1939';

REVOKE ALL ON ALL TABLES IN SCHEMA public FROM Blazkowicz;
REVOKE ALL ON ALL SEQUENCES IN SCHEMA public FROM Blazkowicz;
REVOKE ALL ON ALL FUNCTIONS IN SCHEMA public FROM Blazkowicz;

GRANT CONNECT ON DATABASE banking_system TO Blazkowicz;
GRANT USAGE ON SCHEMA public TO Blazkowicz;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO Blazkowicz;

ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT SELECT ON TABLES TO Blazkowicz;
CREATE ROLE
REVOKE
REVOKE
REVOKE
GRANT
GRANT
GRANT
ALTER DEFAULT PRIVILEGES
banking_system=# \du Blazkowicz
      List of roles
 Role name | Attributes | Member of
-----+-----+-----+
blazkowicz |          | { }
```

### قسمت ششم:

تریگرها:

```
functions.sql
1 CREATE OR REPLACE FUNCTION log_account_creation_date()
2 RETURNS TRIGGER AS $$ 
3 BEGIN
4     NEW.open_date := CURRENT_DATE;
5     RETURN NEW;
6 END;
7 $$ LANGUAGE plpgsql;
8
9 CREATE TRIGGER trg_log_account_creation_date
10 BEFORE INSERT ON account_details
11 FOR EACH ROW
12 EXECUTE FUNCTION log_account_creation_date();
13
14 CREATE OR REPLACE FUNCTION prevent_customer_deletion_with_loans()
15 RETURNS TRIGGER AS $$ 
16 BEGIN
17     IF EXISTS (
18         SELECT 1
19         FROM loan
20         WHERE customer_id = OLD.customer_id
21         AND loan_status = 'Active'
22     ) THEN
23         RAISE EXCEPTION 'cannot delete customer with active loans.';
24     END IF;
25     RETURN OLD;
26 END;
27 $$ LANGUAGE plpgsql;
28
29 CREATE TRIGGER trg_prevent_customer_deletion
30 BEFORE DELETE ON customer_details
31 FOR EACH ROW
32 EXECUTE FUNCTION prevent_customer_deletion_with_loans();
33
34 CREATE OR REPLACE FUNCTION update_account_balance_after_transaction()
35 RETURNS TRIGGER AS $$ 
36 BEGIN
37     IF TG_OP = 'INSERT' THEN
38         IF NEW.transaction_type IN ('Withdrawal', 'Transfer') THEN
39             UPDATE account_details
40                 SET balance = balance - NEW.amount
41                 WHERE account_id = NEW.source_account_id;
42         END IF;
43
44         IF NEW.transaction_type IN ('Deposit', 'Transfer') AND NEW.destination_account_id IS NOT NULL THEN
45             UPDATE account_details
46                 SET balance = balance + NEW.amount
47                 WHERE account_id = NEW.destination_account_id;
48         END IF;
49     END IF;
50
51     RETURN NEW;
52 END;
53 $$ LANGUAGE plpgsql;
54
55 CREATE TRIGGER trg_update_account_balance
56 AFTER INSERT ON transaction
57 FOR EACH ROW
58 EXECUTE FUNCTION update_account_balance_after_transaction();
59
60 CREATE OR REPLACE FUNCTION check_sufficient_balance()
61 RETURNS TRIGGER AS $$ 
62 BEGIN
63     IF NEW.transaction_type IN ('Withdrawal', 'Transfer') THEN
64         PERFORM balance
65             FROM account_details
66             WHERE account_id = NEW.source_account_id
67             AND balance >= NEW.amount;
68
69         IF NOT FOUND THEN
70             RAISE EXCEPTION 'insufficient balance for this transaction.';
71         END IF;
72     END IF;
73
74     RETURN NEW;
75 END;
76 $$ LANGUAGE plpgsql;
77
78 CREATE TRIGGER trg_check_sufficient_balance
79 BEFORE INSERT ON transaction
80 FOR EACH ROW
81 EXECUTE FUNCTION check_sufficient_balance();
82
```

## فانکشن ها:

```
83 CREATE OR REPLACE FUNCTION calculate_total_balance(customer_id INTEGER)
84 RETURNS DECIMAL(15, 2) AS $$ 
85 DECLARE
86     total_balance DECIMAL(15, 2);
87 BEGIN
88     SELECT COALESCE(SUM(balance), 0)
89     INTO total_balance
90     FROM account_details
91     WHERE customer_id = customer_id;
92
93     RETURN total_balance;
94 END;
95 $$ LANGUAGE plpgsql;
96
97 CREATE OR REPLACE FUNCTION check_loan_status(loan_id INTEGER)
98 RETURNS TEXT AS $$ 
99 DECLARE
100    loan_status TEXT;
101 BEGIN
102     SELECT CASE
103         WHEN end_date <= CURRENT_DATE THEN 'Settled'
104         ELSE loan_status
105     END
106     INTO loan_status
107     FROM loan
108     WHERE loan_id = loan_id;
109
110     RETURN loan_status;
111 END;
112 $$ LANGUAGE plpgsql;
113
114 CREATE OR REPLACE FUNCTION count_active_loans(customer_id INTEGER)
115 RETURNS INTEGER AS $$ 
116 DECLARE
117     active_loan_count INTEGER;
118 BEGIN
119     SELECT COUNT(*)
120     INTO active_loan_count
121     FROM loan
122     WHERE customer_id = customer_id
123     | AND loan_status = 'Active';
124
125     RETURN active_loan_count;
126 END;
127 $$ LANGUAGE plpgsql;
128
129 CREATE OR REPLACE FUNCTION calculate_total_loan_payments(loan_id INTEGER)
130 RETURNS DECIMAL(15, 2) AS $$ 
131 DECLARE
132     total_payments DECIMAL(15, 2);
133 BEGIN
134     SELECT COALESCE(SUM(payment_amount), 0)
135     INTO total_payments
136     FROM loanPayment
137     WHERE loan_id = loan_id;
138
139     RETURN total_payments;
140 END;
141 $$ LANGUAGE plpgsql;
142
143 CREATE OR REPLACE FUNCTION get_customer_name(customer_id INTEGER)
144 RETURNS TEXT AS $$ 
145 DECLARE
146     customer_name TEXT;
147 BEGIN
148     SELECT CONCAT(first_name, ' ', last_name)
149     INTO customer_name
150     FROM customer_details
151     WHERE customer_id = customer_id;
152
153     RETURN customer_name;
154 END.
```