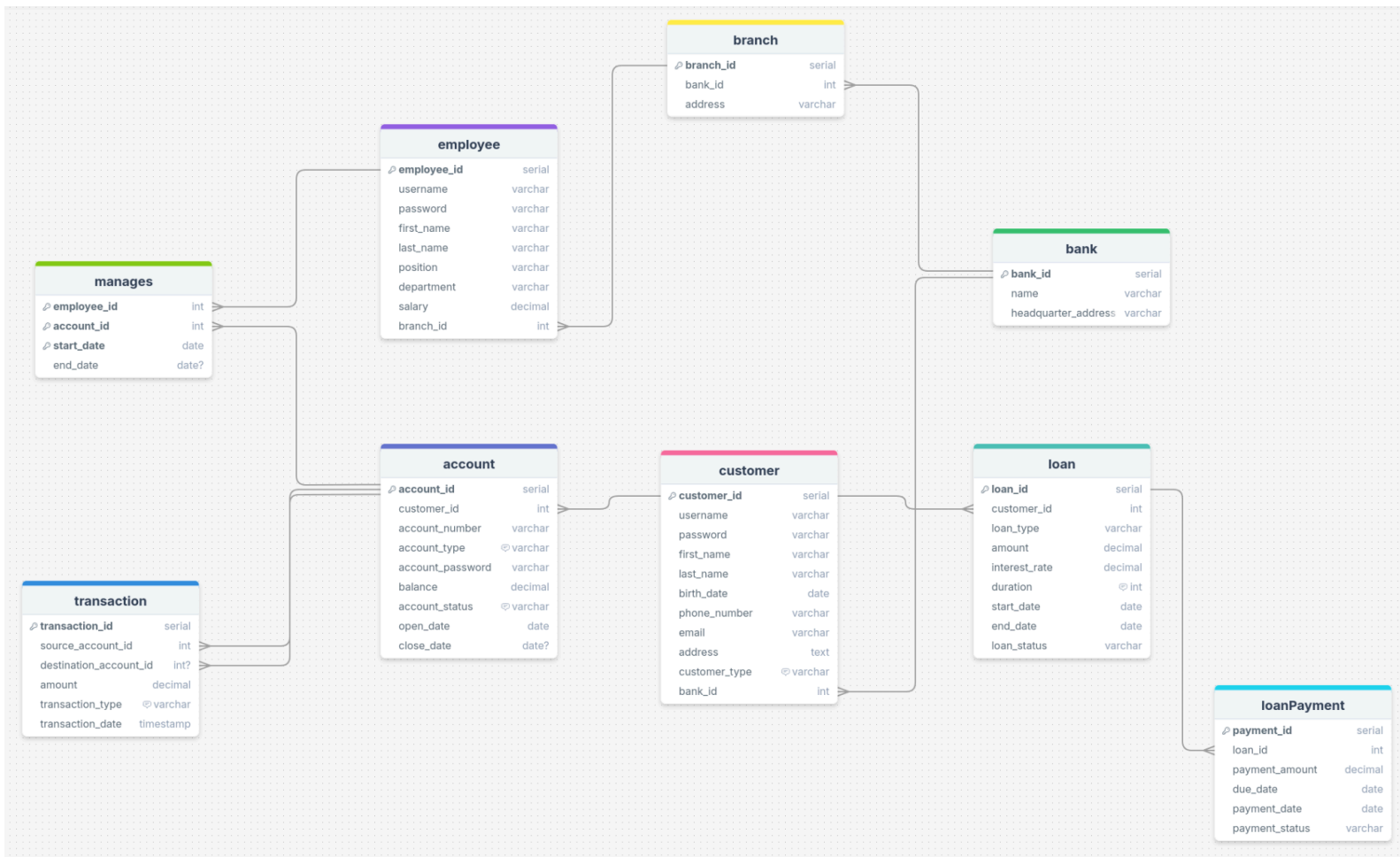


به نام خدا

ندا زارعی - ۴۰۰۲۳۰۳۵

گزارش فاز اول پروژه پایگاه داده

قسمت اول) طراحی اسکیم:



جدول bank:

bank_id (کلید اصلی)

این جدول اطلاعات مربوط به بانک را نگهداری می‌کند

ویژگی‌ها: bank_id (شناسه بانک) و headquarter_address (آدرس مرکزی)

ارتباط این جدول با جدول branch یک به چند است؛ یعنی هر بانک ممکن است چندین شعبه داشته باشد

جدول branch:

branch_id (کلید اصلی)، bank_id (کلید خارجی)

در این جدول اطلاعات مربوط به شعب بانک ذخیره می‌شود

ویژگی‌ها: branch_id (شناسه شعبه)، bank_id (شناسه بانک)، address (آدرس)

ارتباط این جدول با جدول employee یک به چند است؛ یعنی هر شعبه ممکن است چندین کارمند داشته باشد. همچنین ارتباط این جدول با جدول bank به یک به چند است؛ زیرا هر بانک ممکن است چندین شعبه داشته باشد

جدول employee:

employee_id (کلید اصلی)، branch_id (کلید خارجی)

این جدول اطلاعات مربوط به کارمندان بانک را نگهداری می‌کند

ویژگی‌ها: employee_id (شناسه کارمند)، first_name (نام)، last_name (نام خانوادگی)، position (سمت)، department (دپارتمان)، salary (حقوق) و branch_id (شناسه شعبه)

ارتباط این جدول با جدول branch یک به چند است؛ یعنی هر شعبه ممکن است چندین کارمند داشته باشد.

جدول account:

account_id (کلید اصلی)

در این جدول اطلاعات مربوط به حساب‌های بانکی ذخیره می‌شود.

ویژگی‌ها: account_id (شناسه حساب)، account_number (شماره حساب)، account_type (نوع حساب)، balance (موجودی)، account_status (وضعیت حساب)، open_date (تاریخ بازکردن حساب) و close_date (تاریخ بستن حساب)

ارتباط این جدول با جدول customer یک به چند است؛ یعنی هر مشتری ممکن است چندین حساب داشته باشد

جدول manages:

کلید اصلی: (employee_id, account_id, start_date)

هر کارمند می‌تونه چندین اکانت رو مدیریت کنه و هر اکانت می‌تونه توسط چندین کارمند مختلف مورد بررسی و مدیریت قرار بگیره. از اونجایی ارتباط چند به چند دارند باید یک جدول میانی بینشان قرار بگیرد که همان manages است.

این جدول ارتباط بین کارمندان و حساب‌هایی که مدیریت می‌کنند رو نشون میده. ویژگی‌هاش شامل شناسه کارمند، شناسه حساب، تاریخ شروع مدیریت و تاریخ پایان مدیریتته

این جدول ارتباط بسیار مهمی رو بین کارمندان و حساب‌ها نشون میده. هر کارمند می‌تونه چند حساب رو مدیریت کنه و هر حساب هم می‌تونه توسط چند کارمند مدیریت بشه، اما در زمان‌های متفاوت. این جزییات مدیریت حساب‌ها رو ثبت می‌کنه

جدول customer:

این جدول اطلاعات مربوط به مشتریان بانک را نگهداری می‌کند

ویژگی‌ها: `customer_id` (شناسه مشتری)، `first_name` (نام)، `last_name` (نام خانوادگی)، `birth_date` (تاریخ تولد)، `phone_number` (شماره تلفن)، `email` (ایمیل) و `address` (آدرس)

ارتباط این جدول با جدول `account` یک به چند است؛ یعنی هر مشتری ممکن است چندین حساب داشته باشد. همچنین ارتباط این جدول با جدول `loan` به صورت یک به چند است؛ زیرا هر مشتری ممکن است چندین وام دریافت کرده باشد

جدول `loan`:

این جدول اطلاعات مربوط به وام‌های ارائه شده به مشتریان را نگهداری می‌کند.

ویژگی‌ها: `loan_id` (شناسه وام)، `customer_id` (شناسه مشتری)، `loan_type` (نوع وام)، `amount` (مبلغ)، `interest_rate` (نرخ بهره)، `duration` (مدت)، `start_date` (تاریخ شروع)، `end_date` (تاریخ پایان) و `loan_status` (وضعیت وام)

ارتباط این جدول با جدول `customer` یک به چند است؛ یعنی هر مشتری ممکن است چندین وام دریافت کرده باشد.

ارتباط این جدول با جدول `loanPayment` به صورت یک به چند است؛ زیرا هر وام ممکن است شامل چندین پرداخت باشد.

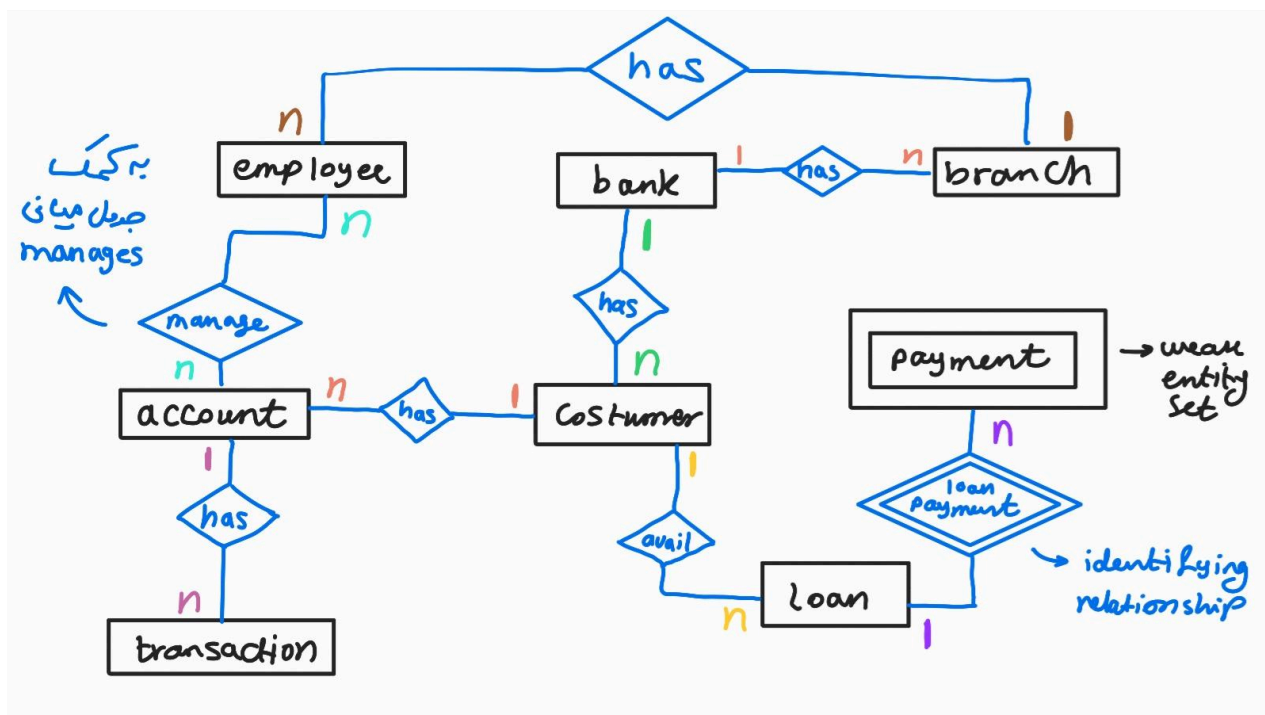
جدول `loanPayment`:

این جدول اطلاعات مربوط به پرداخت‌های انجام شده (اقساط) برای وام‌ها را نگهداری می‌کند

ویژگی‌ها: `payment_id` (شناسه پرداخت)، `loan_id` (شناسه وام)، `payment_amount` (مبلغ پرداخت)، `due_date` (تاریخ سررسید) و `payment_date` (تاریخ پرداخت).

ارتباط این جدول با جدول `loan` یک به چند است؛ زیرا هر وام ممکن است شامل چندین پرداخت یا قسط باشد

قسمت دوم) نمودار ER :



قسمت سوم) نرمال سازی BCNF:

جدول bank:

کلید اصلی: bank_id

وابستگی‌ها:

$bank_id \rightarrow headquarter_address$

جدول در BCNF است چون تنها وابستگی تابعی، کلید اصلی را شامل می‌شود

جدول branch:

کلید اصلی: branch_id

وابستگی‌ها:

$branch_id \rightarrow bank_id, address, manager_id$

$bank_id \rightarrow branch_id$ (ارتباط با بانک)

این جدول در BCNF است چون تمام وابستگی‌های تابعی به کلید اصلی وابسته هستند

3. جدول employee:

کلید اصلی: employee_id

وابستگی‌ها:

$employee_id \rightarrow first_name, last_name, position, department, salary, branch_id$

جدول در BCNF است زیرا وابستگی تابعی تنها از کلید اصلی به دیگر ویژگی‌ها است.

4. جدول account:

کلید اصلی: account_id

وابستگی‌ها:

account_id → account_number, account_type, balance, account_status, open_date, close_date

جدول در BCNF است زیرا تمام وابستگی‌ها بر اساس کلید اصلی هستند.

5. جدول manages:

کلید اصلی: ترکیب employee_id و account_id و start_date

وابستگی‌ها:

employee_id, account_id, start_date → end_date

جدول در BCNF است زیرا ترکیب کلید اصلی تمام وابستگی‌های تابعی را پوشش می‌دهد

6. جدول customer:

کلید اصلی: customer_id

وابستگی‌ها:

customer_id → first_name, last_name, birth_date, phone_number, email, address

جدول در BCNF است

7. جدول loan:

کلید اصلی: loan_id

وابستگی‌ها:

loan_id → customer_id, loan_type, amount, interest_rate, duration, start_date, end_date,
loan_status

جدول در BCNF است

8. جدول loanPayment:

کلید اصلی: payment_id

وابستگی‌ها:

$\text{payment_id} \rightarrow \text{loan_id}, \text{payment_amount}, \text{due_date}, \text{payment_date}$

جدول در BCNF است زیرا تنها وابستگی تابعی از کلید اصلی به دیگر ویژگی‌ها وجود دارد

پس جدول خودش نرمال BCNF است و نیازی به تغییرش نیست.

```

docker-compose.yml M  Dockerfile  database.go M  transaction.go U
main.go > main
func main() {
    if err := db.Connect(); err != nil {
        log.Fatal(err)
    }
    fmt.Println("connected to database")

    bank := &model.Bank{
        Name: "Sepah Bank",
        HeadquarterAddress: "123 Main St",
    }
    db.CreateBank(bank)

    branch := &model.Branch{
        BankID: 1,
        Address: "456 Elm St",
    }
    db.CreateBranch(branch)

    employee := &model.Employee{
        Username: "rossbenson",
        Password: "password",
        FirstName: "ross",
        LastName: "benson",
        Position: "teller",
        Department: "operations",
        Salary: 50000,
        BranchID: 1,
    }
    db.RegisterEmployee(employee)

    customer := &model.Customer{
        Username: "eddiebooker",
        Password: "password2627",
        FirstName: "eddie",
        LastName: "booker",
        BirthDate: "1990-01-01",
        PhoneNumber: "1234567890",
        Email: "eddiebooker@gmail.com",
        Address: "3202 Trails end road",
        CustomerType: "Individual",
        BankID: 1,
    }
    db.RegisterCustomer(customer)

    customer, err := db.LoginCustomer("eddiebooker", "password2627")
    if err != nil {
        log.Fatal(err)
    }
    fmt.Printf("logged in customer: %v\n", customer)

    account := &model.Account{
        AccountNumber: "1234567890",
        AccountType: "Savings",
        AccountPassword: "passpass",
        Balance: 1000,
        AccountStatus: "Active",
        OpenDate: "2025-01-01",
        CustomerID: 1,
    }
    db.CreateAccount(account)

    fmt.Println("getting the customer 1 account:")
    db.GetAccount("1234567890", "passpass")

    transaction := &model.Transaction{
        SourceAccountID: 1,
        DestinationAccountID: nil,
        Amount: 1000,
        TransactionType: "Deposit",
        TransactionDate: "2025-01-01",
    }
    db.CreateTransaction(transaction)
}

```

```
postgres | 2024-12-13 18:35:20.552 UTC [45] LOG:  database system was shut down at 2024-12-13 18:35:20.552 UTC
postgres | 2024-12-13 18:35:20.560 UTC [48] LOG:  database system is ready to accept connections
postgres | done
postgres | server started
postgres | CREATE DATABASE
postgres |
postgres | /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/tables.sql
postgres | CREATE TABLE
postgres | CREATE TABLE
postgres | CREATE TABLE
postgres | CREATE TABLE
postgres | CREATE TABLE
postgres | COMMENT
postgres | CREATE TABLE
postgres | COMMENT
postgres | CREATE TABLE
postgres | COMMENT
postgres | CREATE TABLE
postgres | COMMENT
postgres | CREATE TABLE
postgres |
```

```
app | connected to database
app | successfully created bank
app | successfully created branch
app | successfully registered employee
app | successfully registered
app | logged in customer: &{CustomerID:1 Username:eddiebooker Password:password2627 FirstName:eddie LastName:booker BirthDate:1990-01-01T00:00:00Z PhoneNumber:1234567890 Email:eddiebooker@gmail.com Address:3202 Trails end road CustomerType:Individual BankID:1}
app | successfully created account
app | getting the costumer 1 account:
app | successfully created transaction
app | Starting server on :8080
```

```
 account_id | customer_id | account_number | account_type | account_password | balance | account_status | open_date | close_date
-----+-----+-----+-----+-----+-----+-----+-----+-----
      1 |          1 | 1234567890 | Savings | passpass | 1000.00 | Active | 2025-01-01 |
(1 row)
```

```
customer_id | username | password | first_name | last_name | birth_date | phone_number | email | address | customer_type | bank_id
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
      1 | eddiebooker | password2627 | eddie | booker | 1990-01-01 | 1234567890 | eddiebooker@gmail.com | 3202 Trails end road | Individual | 1
(1 row)
```

```
banking_system=# select * from transaction;
transaction_id | source_account_id | destination_account_id | amount | transaction_type | transaction_date
-----+-----+-----+-----+-----+-----
      1 |          1 |          1 | 1000.00 | Deposit | 2025-01-01 00:00:00
(1 row)

banking_system=#
```