

Name: Sarthak Shandilya

Coding Challenge 5 (Loan Management)

Submitted to: Karthika

Create SQL Schema from the customer and loan class, use the class attributes for table column names.

1. Define a `Customer` class with the following confidential attributes:

- a. Customer ID
- b. Name
- c. Email Address
- d. Phone Number
- e. Address
- f. creditScore

```
mysql> create table customers(  
  -> customer_id int primary key auto_increment,  
  -> name varchar(30),  
  -> email varchar(40),  
  -> phone long,  
  -> address text,  
  -> creditScore int);  
Query OK, 0 rows affected (0.39 sec)
```

2. Define a base class `Loan` with the following attributes:

- a. loanId
- b. customer (reference of customer class)
- c. principalAmount
- d. interestRate
- e. loanTerm (Loan Tenure in months)
- f. loanType (CarLoan, HomeLoan)
- g. loanStatus (Pending, Approved)

```
mysql> create table loan(  
-> loan_id int primary key auto_increment,  
-> customer_id int,  
-> principal_amount float,  
-> interest_rate float,  
-> loan_term int,  
-> loan_type enum('HomeLoan', 'CarLoan'),  
-> loan_status enum('Pending', 'Approved'),  
-> foreign key(customer_id) references customers(customer_id) on delete  
cascade on update cascade);  
Query OK, 0 rows affected (0.22 sec)
```

3. Create two subclasses: `HomeLoan` and `CarLoan`. These subclasses should inherit from the Loan class and add attributes specific to their loan types.

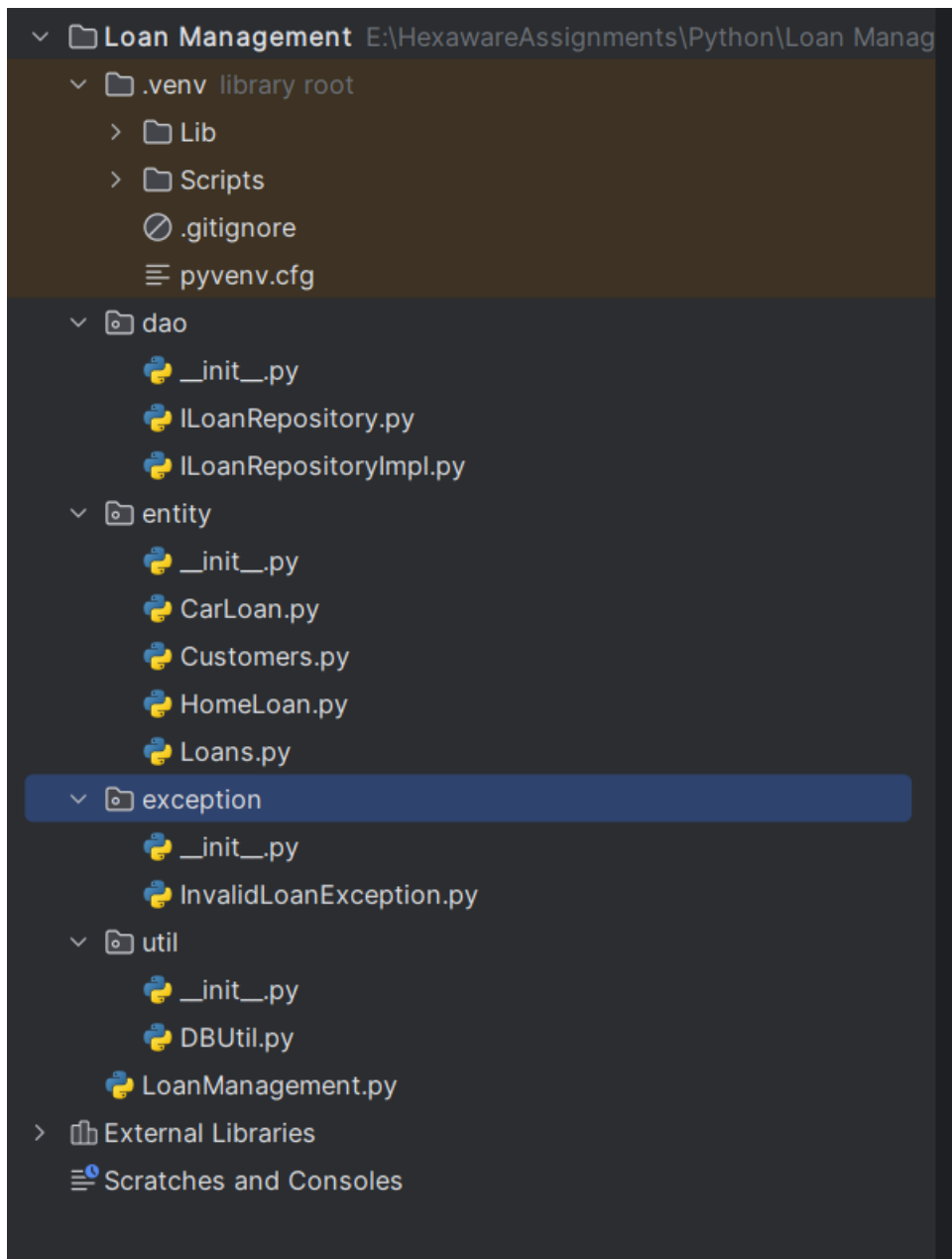
For example:

- a. HomeLoan should have a propertyAddress (String) and propertyValue (int) attribute.

```
mysql> create table HomeLoan(  
-> loan_id int primary key,  
-> propertyAddress varchar(255),  
-> propertyValue decimal(10,2),  
-> foreign key(loan_id) references loan(loan_id) on delete cascade on update cascade);  
Query OK, 0 rows affected (0.06 sec)
```

- b. CarLoan should have a carModel (String) and carValue (int) attribute.

```
mysql> create table CarLoan(  
-> loan_id int primary key,  
-> carModel varchar(50),  
-> carValue decimal(10,2),  
-> foreign key(loan_id) references loan(loan_id) on delete cascade on update cascade);  
Query OK, 0 rows affected (0.09 sec)
```



This is the packages and project structure.

Below are the classes implementations:

Customers:

```
1
2 3 usages
3 class Customer:
4     def __init__(self, name, email, phone, address, creditScore):
5         self.customer_id = None
6         self.name = name
7         self.email = email
8         self.phone = phone
9         self.address = address
10        self.creditScore = creditScore
11
12 1 usage
13 @property
14 def customerId(self):
15     return self.customer_id
16
17 2 usages
18 @customerId.setter
19 def customerId(self, customer_id):
20     self.customer_id = customer_id
```

Loans:

```
1 8 usages
2 class Loan:
3     def __init__(self, customer, principal_amount, interest_rate, loan_term, loan_type, loan_status):
4         self.customer = customer
5         self.loan_id = None
6         self.principal_amount = principal_amount
7         self.interest = interest_rate
8         self.loan_term = loan_term
9         self.loan_type = loan_type
10        self.loan_status = loan_status
11
12 1 usage
13 @property
14 def loanId(self):
15     return self.loan_id
16
17 @loanId.setter
18 def loanId(self, loan_id):
19     self.loan_id = loan_id
```

Home Loan:

```
1 from Loans import Loan
2
3
4 4 usages
5 class HomeLoan(Loan):
6     def __init__(self, customer, principal_amount, interest_rate, loan_term, loan_type, loan_status, address, value):
7         super().__init__(customer, principal_amount, interest_rate, loan_term, loan_type, loan_status)
8         self.property_address = address
9         self.property_value = value
```

Car Loan:

```
from Loans import Loan

4 usages
class CarLoan(Loan):
    def __init__(self, customer, principal_amount, interest_rate, loan_term, loan_type, loan_status, car_model, car_value):
        super().__init__(customer, principal_amount, interest_rate, loan_term, loan_type, loan_status)
        self.car_model = car_model
        self.car_value = car_value
```

Interface ILoanRepository:

```
from entity.Loans import Loan

2 usages
class ILoanRepository(ABC):

    @abstractmethod
    def applyLoan(self, loan: Loan):
        pass

    @abstractmethod
    def calculateInterest(self, loanId):
        pass

    @abstractmethod
    def loanStatus(self, loanId):
        pass

    @abstractmethod
    def calculateEMI(self, loanId):
        pass

    @abstractmethod
    def loanRepayment(self, loanId, amount):
        pass

    @abstractmethod
    def getAllLoan(self):
        pass
```

```
@abstractmethod
def getLoanById(self, loanId):
    pass
```

Implementation of ILoanRepositoryImpl:

```
from entity.CarLoan import CarLoan
from entity.Customers import Customer
from util.DBUtil import DBUtil
from entity.HomeLoan import HomeLoan
from dao.ILoanRepository import ILoanRepository
from exception.InvalidLoanException import InvalidLoanException
from entity.Loans import Loan

2 usages
class ILoanRepositoryImpl(ILoanRepository):
    def __init__(self):
        super().__init__()
        self.con = DBUtil.getDBConn()

1 usage
def get_customer(self):
    print("Are you a new customer?")
    choice = input("Enter yes or no : ").lower()
    if choice == "no":
        customer_id = int(input("Enter your customer id : "))
        cursor = self.con.cursor()
        cursor.execute("select * from customers where customer_id = %s", (customer_id,))
        customer_data = cursor.fetchone()
        if customer_data:
            name = customer_data[1]
            email = customer_data[2]
            phone = customer_data[3]
            address = customer_data[4]
            creditscore = customer_data[5]
            customer = Customer(name, email, phone, address, creditscore)
```

```
        address = customer_data[4]
        creditscore = customer_data[5]
        customer = Customer(name, email, phone, address, creditscore)
        customer.customerId = customer_id
        return customer

    elif "yes":
        name = input("Enter your name : ")
        email = input("Enter your email : ")
        phone = input("Enter your phone : ")
        address = input("Enter your address : ")
        creditScore = int(input("Enter your credit score : "))
        cursor = self.con.cursor()
        q = "insert into customers(name,email,phone,address,creditScore) values (%s,%s,%s,%s,%s)"
        cursor.execute(q, (name, email, phone, address, creditScore,))
        self.con.commit()
        customer_id = cursor.lastrowid
        customer = Customer(name, email, phone, address, creditScore)
        customer.customerId = customer_id
        print("You're successfully registered as a new customer.")
        print(f"Your customer id is {customer_id}")
        return customer

1 usage
def applyLoan(self, loan: Loan):
    confirm = input("Please confirm if you want to apply for loan (Yes/No)").lower()
    if confirm == "yes":
        cursor = self.con.cursor()
        query = "insert into loan (customer_id,principal_amount,interest_rate,loan_term,loan_type,loan_status"
        values = (loan.customer.customer_id, loan.principal_amount, loan.interest, loan.loan_term, loan.loan_
        loan.loan_status)
        cursor.execute(query, values)
```

applyLoan():

```
1 usage
def applyLoan(self, loan: Loan):
    confirm = input("Please confirm if you want to apply for loan (Yes/No)").lower()
    if confirm == "yes":
        cursor = self.con.cursor()
        query = "insert into loan (customer_id,principal_amount,interest_rate,loan_term,loan_type,loan_status"
        values = (loan.customer.customer_id, loan.principal_amount, loan.interest, loan.loan_term, loan.loan_
        loan.loan_status)
        cursor.execute(query, values)
        self.con.commit()
        loan.loan_id = cursor.lastrowid
        loan_id = loan.loan_id
        if isinstance(loan, HomeLoan):
            q_home = "insert into HomeLoan (loan_id,propertyAddress,propertyValue) values (%s,%s,%s)"
            loanData = (loan_id, loan.property_address, loan.property_value)
            cursor.execute(q_home, loanData)
            self.con.commit()
        elif isinstance(loan, CarLoan):
            q_car = "insert into CarLoan (loan_id, carModel, carValue) values (%s,%s,%s)"
            loanData = (loan_id, loan.car_model, loan.car_value)
            cursor.execute(q_car, loanData)
            self.con.commit()
        print("Congratulations! You've successfully applied for the loan.")
        print(f"Your loan id is {loan_id}")
    else:
        print("Loan application cancelled.")
```

calculateInterest() and loanStatus():


```

1 usage
def calculateInterest(self, loanId):
    cursor = self.con.cursor()
    cursor.execute("select interest_rate from loan where loan_id=%s", (loanId,))
    monthly_interest = cursor.fetchone()
    if monthly_interest:
        return monthly_interest[0]/1200
    else:
        raise InvalidLoanException("Loan not found")

1 usage
def loanStatus(self, loanId):
    cursor = self.con.cursor()
    cursor.execute("select creditScore from customers join loan on loan.customer_id=customers.customer_id and loan_id = %s", (loanId,))
    stat = cursor.fetchone()
    if stat:
        credScore = stat[0]
        if credScore > 650:
            q = "update loan set loan_status='Approved' where loan_id=%s"
            cursor.execute(q, (loanId,))
            self.con.commit()
            print("Congratulations! Your loan is Approved.")
        else:
            q = "update loan set loan_status='Rejected' where loan_id=%s"
            cursor.execute(q, (loanId,))
            self.con.commit()
            print("Sorry! Your loan is rejected due to low credit score.")
    else:
        raise InvalidLoanException("Loan not found.")

```

CalculateEMI() and LoanRepayment():

```

def calculateEMI(self, loanId):
    cursor = self.con.cursor()
    query = "select principal_amount,interest_rate,loan_term from loan where loan_id=%s"
    cursor.execute(query, (loanId,))
    loan_data = cursor.fetchone()
    if loan_data:
        principal, interest_annual, term = loan_data
        interest = self.calculateInterest(loanId)
        emi = (principal*interest*(1+interest)**term)/(((1+interest)**term)-1)
        return emi
    else:
        InvalidLoanException("Loan not exists.")

1 usage
def loanRepayment(self, loanId, amount):
    cursor = self.con.cursor()
    query = "select principal_amount,interest_rate,loan_term from loan where loan_id=%s"
    cursor.execute(query, (loanId,))
    loan_data = cursor.fetchone()
    if loan_data:
        principal, interest, loanTerm = loan_data
        emi = self.calculateEMI(loanId)
        emi_possible = amount//emi
        if emi_possible < 1:
            print("Payment Rejected. Amount is less than a single EMI. ")
        else:
            print(f"{int(emi_possible)} EMI's paid from the amount.")
    else:
        raise InvalidLoanException("Loan Not Found.")
    print("We're heading you to main menu.")

```

getAllLoan() and getLoanById():

```
1 usage
def getAllLoan(self):
    cursor = self.con.cursor()
    cursor.execute("select * from loan")
    return cursor.fetchall()

1 usage
def getLoanById(self, loanId):
    cursor = self.con.cursor()
    cursor.execute("select * from loan where loan_id=%s", (loanId,))
    return cursor.fetchall()
```

LoanManagement:

```
1 from entity.CarLoan import CarLoan
2 from entity.HomeLoan import HomeLoan
3 from dao.ILoanRepositoryImpl import ILoanRepositoryImpl
4
5
6 1 usage
7 class LoanManagement(ILoanRepositoryImpl):
8     def __init__(self):
9         super().__init__()
10
11 1 usage
12 def main(self):
13     while True:
14         print("----Menu----")
15         print("1. Apply loan.")
16         print("2. Get All Loan History.")
17         print("3. Get Your Loan Details.")
18         print("4. Make Loan repayment.")
19         print("5. Get Loan status.")
20         print("6. Exit")
21         choice = int(input("Enter your choice here : "))
22         match choice:
23             case 1:
24                 loan = None
25                 customer = self.get_customer()
26                 loan_amount = float(input("Enter the amount you want to borrow : "))
27                 interestRate = float(input("Enter the interest rate : "))
28                 loanterm = int(input("Enter the tenure for which you want to take loan : "))
29                 loanType = input("Enter your loan type (HomeLoan/CarLoan) : ")
30                 loanstatus = "Pending"
31                 if loanType == "HomeLoan":
```

```

        if loanType == "HomeLoan":
            propertyAddress = input("Enter the address of the property : ")
            propertyValue = float(input("Enter the value of the property : "))
            loan = HomeLoan(customer, loan_amount, interestRate, loanterm, loanType, loanstatus,
                             propertyAddress, propertyValue)
        elif loanType == "CarLoan":
            carModel = input("Enter car model : ")
            carValue = float(input("Enter the value of car : "))
            loan = CarLoan(customer, loan_amount, interestRate, loanterm, loanType, loanstatus, carModel,
                            carValue)
        self.applyLoan(loan)
        print("We're heading you to main menu.")
        print()
    case 2:
        loans = self.getAllLoan()
        for l in loans:
            print("Loan id : ", l[0])
            print("Customer Id : ", l[1])
            print("Principal Amount : ", l[2])
            print("Interest Rate : ", l[3])
            print("Loan term : ", l[4])
            print("Loan type : ", l[5])
            print("Loan status : ", l[6])
    case 3:
        loanId = int(input("Please enter your loan ID : "))
        loans = self.getLoanById(loanId)
        for l in loans:
            print("Loan id : ", l[0])
            print("Customer Id : ", l[1])
            print("Principal Amount : ", l[2])
            print("Interest Rate : ", l[3])

```

```

7         print("Loan status : ", l[6])
8         print("Customer Id : ", l[1])
9         print("Principal Amount : ", l[2])
0         print("Interest Rate : ", l[3])
1         print("Loan term : ", l[4])
2         print("Loan type : ", l[5])
3         print("Loan status : ", l[6])
4
5     case 4:
6         loanId = int(input("Enter your loan Id : "))
7         amount = float(input("Enter the amount you want to repay : "))
8         self.loanRepayment(loanId, amount)
9         print()
0     case 5:
1         loanId = int(input("Enter your loan id : "))
2         self.loanStatus(loanId)
3         print()
4     case 6:
5         print("Thanks for visiting us. Have a good day.")
6         break
7     case _:
8         print("Invalid input please try again.")
9
0 loan = LoanManagement()
1 loan.main()

```

Outputs:

1. Apply Loan (Car/Home) with existing/Not existing customer

```
"E:\HexawareAssignments\Python\Loan Management\.venv\Scripts\python.exe" "E:\HexawareAssignments\Python\Loan Management\LoanManagement.py"
----Menu----
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 1
Are you a new customer?
Enter yes or no : yes
Enter your name : Mayank
Enter your email : mayank@gmail.com
Enter your phone : 9430203375
Enter your address : Aghoria Chowk
Enter your credit score : 750
You're successfully registered as a new customer.
Your customer id is 2
Enter the amount you want to borrow : 54000
Enter the interest rate : 5
Enter the tenure for which you want to take loan : 60
Enter your loan type (HomeLoan/CarLoan) : CarLoan
Enter car model : Verna
Enter the value of car : 200000
Please confirm if you want to apply for loan (Yes/No)Yes
Congratulations! You've successfully applied for the loan.
Your loan id is 2
We're heading you to main menu.
```

```
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 1
Are you a new customer?
Enter yes or no : no
Enter your customer id : 2
Enter the amount you want to borrow : 4000000
Enter the interest rate : 9
Enter the tenure for which you want to take loan : 120
Enter your loan type (HomeLoan/CarLoan) : HomeLoan
Enter the address of the property : Professor colony
Enter the value of the property : 10000000
Please confirm if you want to apply for loan (Yes/No)Yes
Congratulations! You've successfully applied for the loan.
Your loan id is 3
We're heading you to main menu.
```

----Menu----

2. GetAll loan histosy:

```

C:\mezzad\assignments\python\loan_management\env\scripts\python.exe C:\mezzad\assignments\python\loan_management\loandmanagement.py
---Menu---
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 2
Loan id : 1
Customer Id : 1
Principal Amount : 4500000.0
Interest Rate : 9.0
Loan term : 120
Loan type : HomeLoan
Loan status : Pending
-----
Loan id : 2
Customer Id : 2
Principal Amount : 54000.0
Interest Rate : 5.0
Loan term : 60
Loan type : CarLoan
Loan status : Pending
-----
Loan id : 3
Customer Id : 2
Principal Amount : 4000000.0
Interest Rate : 9.0

```

3. Get Loan history by Id

```

-----
---Menu---
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 3
Please enter your loan ID : 3
Loan id : 3
Customer Id : 2
Principal Amount : 4000000.0
Interest Rate : 9.0
Loan term : 120
Loan type : HomeLoan
Loan status : Pending
-----
---Menu---

```

4. Make loan repayment

```
----Menu----
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 4
Enter your loan Id : 2
Enter the amount you want to repay : 10000
9 EMI's paid from the amount.
We're heading you to main menu.
```

5. Get loan status

```
----Menu----
1. Apply loan.
2. Get All Loan History.
3. Get Your Loan Details.
4. Make Loan repayment.
5. Get Loan status.
6. Exit
Enter your choice here : 5
Enter your loan id : 2
Your loan status is Pending
```