**PL/SQL**

**Bank\_Schema :- Created in Oracle SqlDeveloper**

**1. Customers Table:**

**A close-up of a number

AI-generated content may be incorrect.**

**2. Loans Table:**

A white paper with numbers and text

AI-generated content may be incorrect.

**3. Accounts Table:**

A close-up of a paper

AI-generated content may be incorrect.

**4. Transactions Table:**

A close-up of numbers

AI-generated content may be incorrect.

**5. Employees ID:**

A black and white image of a black and white image of a black and white image of a black and white image of a black and white image of a black and white image of a black and

AI-generated content may be incorrect.

**Exercise 1- Control Structure**

1. **Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

**Query:**

BEGIN

FOR cust IN (SELECT customerid, TRUNC(MONTHS\_BETWEEN(SYSDATE, dob)/12) AS age

FROM customers) LOOP

IF cust.age > 60 THEN

UPDATE loans

SET interestrate = interestrate - 1

WHERE customerid = cust.customerid;

END IF;

END LOOP;

COMMIT;

END;

/

**Table after running the above query**

A white paper with numbers and text

AI-generated content may be incorrect.

1. **Scenario 2:** A customer can be promoted to VIP status based on their balance.

**Query:**

BEGIN

FOR cust IN (

SELECT customerid

FROM customers

WHERE balance > 10000

) LOOP

UPDATE customers

SET isvip = 'true'

WHERE customerid = cust.customerid;

END LOOP;

COMMIT;

END;

/

A close-up of a number

AI-generated content may be incorrect. **Table after running the above query**

1. **Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

**Query:**

DECLARE

CURSOR due\_loans IS

SELECT loanid, customerid, enddate

FROM loans where enddate<=sysdate+30;

v\_name customers.name%TYPE;

BEGIN

FOR loan\_rec IN due\_loans LOOP

SELECT name INTO v\_name

FROM customers

WHERE customerid = loan\_rec.customerid ;

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan for ' || v\_name ||' (ID: ' || loan\_rec.customerid || ') is due till ' || TO\_CHAR(loan\_rec.enddate, 'DD-MON-YYYY')

);

END LOOP;

END;

/

**Output:**

A close-up of a computer code

AI-generated content may be incorrect.

**Exercise 3 - Stored Procedures**

1. **Scenario 1:** The bank needs to process monthly interest for all savings accounts.

**Query:**

create procedure processMonthlyInterest is

begin

update accounts

set balance=balance+(balance\*0.01)

where accounttype='Savings';

commit;

end;

/

exec processMonthlyInterest;

**Accounts table After executing the proccessMonthlySalaries Procedure**

A screenshot of a computer screen

AI-generated content may be incorrect.

1. **Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**Query:**

create procedure updateEmployeeBonus(

department in varchar,

bonus\_percent in number)

is

begin

update employees e

set salary=salary+(salary\*bonus\_percent/100)

where e.department=department;

commit;

end;

/

exec updateEmployeeBonus('IT', 5);

**Employee Table after executing updateEmployeeBonus procedure**

**A screenshot of a white background

AI-generated content may be incorrect.**

1. **Scenario 3:** Customers should be able to transfer funds between their accounts.

**Query:**

create procedure Transferfunds(from\_account in number, to\_account in number, amount in number) is

nbalance number;

begin

select balance into nbalance from accounts where accountid = from\_account;

if nbalance<amount then raise\_application\_error(-20001, 'Insufficient balance in source account.');

end if;

update accounts

set balance=balance-amount

where accountid=from\_account;

update accounts

set balance=balance+amount

where accountid=to\_account;

commit;

end;

/

exec Transferfunds(1,2,1050);

**A close-up of a paper

AI-generated content may be incorrect. Accounts Table after executing Transferfunds Procedure**