Islamic University of Technology

CSE-4410
Database Management Systems - II Lab

Lab Report - 4

Name: Mukit Mahdin
ID: 200042170
Department: CSE
Program: SWE

Table Creation and Entry Insertion :

```sql
CREATE TABLE ACCOUNTPROPERTY(
    ID INT PRIMARY KEY,
    NAME VARCHAR2(20),
    PROFITRATE NUMERIC(10,2),
    GRACEPERIOD INT
);

CREATE TABLE ACCOUNT(
    ID INT PRIMARY KEY,
    NAME VARCHAR2(20),
    ACCCODE INT,
    OPENINGDATE DATE,
    LASTDATEINTEREST DATE,

    CONSTRAINT FK_ACCOUNT_ACCOUNTPROPERTY FOREIGN KEY(ACCCODE) REFERENCES
ACCOUNTPROPERTY(ID)
);

CREATE TABLE TRANSACTION(
    TID INT PRIMARY KEY,
    ACCNO INT,
    AMOUNT NUMERIC(10,2),
    TRANSACTIONDATE DATE,

    CONSTRAINT FK_TRANSACTION_ACCNO FOREIGN KEY(ACCNO) REFERENCES
ACCOUNT(ID)
);

CREATE TABLE BALANCE(
    ACCNO INT PRIMARY KEY,
    PRINCIPALAMOUNT BALANCE NUMERIC(10,2),
    PRINCIPALAMOUNT PROFIT NUMERIC(10,2),

    CONSTRAINT FK_BALANCE_ACCNO FOREIGN KEY(ACCNO) REFERENCES ACCOUNT(ID)
);

INSERT INTO ACCOUNT VALUES(1,'A',1,'01-01-2017','01-01-2018');
INSERT INTO ACCOUNT VALUES(2,'B',2,'01-01-2017','01-02-2018');
INSERT INTO ACCOUNT VALUES(3,'C',3,'01-01-2017','01-03-2018');
INSERT INTO ACCOUNT VALUES(4,'D',4,'01-01-2017','01-04-2018');
INSERT INTO ACCOUNT VALUES(5,'E',5,'01-01-2017','01-05-2018');

INSERT INTO ACCOUNTPROPERTY VALUES(1,'A',1,1);
INSERT INTO ACCOUNTPROPERTY VALUES(2,'B',2,2);
INSERT INTO ACCOUNTPROPERTY VALUES(3,'C',3,3);
INSERT INTO ACCOUNTPROPERTY VALUES(4,'D',4,4);
INSERT INTO ACCOUNTPROPERTY VALUES(5,'E',5,5);
INSERT INTO BALANCE VALUES(1,1000,1000);
INSERT INTO BALANCE VALUES(2,2000,2000);
INSERT INTO BALANCE VALUES(3,3000,3000);
INSERT INTO BALANCE VALUES(4,4000,4000);
INSERT INTO BALANCE VALUES(5,5000,5000);

INSERT INTO TRANSACTION VALUES(1,1,1000,'01-01-2018');
INSERT INTO TRANSACTION VALUES(2,2,2000,'01-02-2018');
INSERT INTO TRANSACTION VALUES(3,3,3000,'01-03-2018');
INSERT INTO TRANSACTION VALUES(4,4,4000,'01-04-2018');
INSERT INTO TRANSACTION VALUES(5,5,5000,'01-05-2018');
```

Task - 1 :

You have to write a function to calculate the current balance from the transactions.

```
CREATE OR REPLACE FUNCTION calculate_balance (p_accno INT)
RETURN NUMERIC
IS
  v_balance NUMERIC(10, 2);
BEGIN
  SELECT SUM(AMOUNT)
  INTO v_balance
  FROM TRANSACTION
  WHERE ACCNO = p_accno;

  UPDATE BALANCE
  SET PRINCIPALAMOUNT = PRINCIPALAMOUNT + v_balance,
      PROFITAMOUNT = PROFITAMOUNT + v_balance * (SELECT PROFITRATE FROM
ACCOUNTPROPERTY WHERE ID =
                       (SELECT ACCCODE FROM ACCOUNT WHERE ID = p_accno))
  WHERE ACCNO = p_accno;

  SELECT PRINCIPALAMOUNT + PROFITAMOUNT
  INTO v_balance
  FROM BALANCE
  WHERE ACCNO = p_accno;

  RETURN v_balance;
END;
/
```

Description :

The function "calculate_balance" that takes an account number (p_accno) as input and returns the balance of that account as a numeric value.

The function starts by selecting the sum of all transactions associated with the account number (p_accno) from the "TRANSACTION" table and storing it in a variable "v_balance".

Next, the function updates the balance for that account in the "BALANCE" table by adding the transaction sum (v_balance) to the current balance and also calculates the profit based on the profit rate from the "ACCOUNTPROPERTY" table and adds that to the balance as well. The profit rate is retrieved by joining the "ACCOUNT" and "ACCOUNTPROPERTY" tables using the account code.

Finally, the function selects the updated balance (principal amount + profit amount) from the "BALANCE" table and stores it in the "v_balance" variable. The function returns the value of "v_balance" as the final output.

Task - 2 :

Write another function to calculate the profit based on profitRate, amount and duration. Take account id as input and return profit, balance before profit, and balance after profit.

```
CREATE OR REPLACE FUNCTION CALCULATE_PROFIT(ACCNO INT)
RETURN NUMERIC
IS
  BALANCE NUMERIC(10,2);
  PROFIT NUMERIC(10,2);
  PROFIT_RATE NUMERIC(10,2);
  PRINCIPAL NUMERIC(10,2);
  OPENING_DATE DATE;
  DURATION INT;

BEGIN
  SELECT SUM(AMOUNT), PROFITRATE, OPENINGDATE
  INTO BALANCE,PROFIT_RATE,OPENING_DATE
  FROM TRANSACTION
  JOIN ACCOUNT ON ACCOUNT.ID = TRANSACTION.ACCNO
  JOIN ACCOUNTPROPERTY ON ACCOUNTPROPERTY.ID = ACCOUNT.ACCCODE
  WHERE ACCNO = ACCNO;

  DURATION := ROUND((SYSDATE - OPENING_DATE));
  PROFIT := BALANCE*PROFIT_RATE*DURATION/365;
  PRINCIPAL := BALANCE - PROFIT;
  RETURN PROFIT,PRINCIPAL,BALANCE;
END;
/
```

Description :

The above code defines a PL/SQL function called "CALCULATE_PROFIT". This function takes a single input parameter "ACCNO", which represents the account number.

The function returns a numeric value, which is the profit calculated for the account.

The function first selects the sum of amounts, profit rate, and opening date for the transactions of the account number passed as the parameter. This information is obtained by joining the "TRANSACTION", "ACCOUNT", and "ACCOUNTPROPERTY" tables using the account number.

Next, the duration of the account is calculated by subtracting the opening date from the current system date and rounding the result.

Then, the profit is calculated using the formula: balance * profit rate * duration / 365. The principal amount is calculated by subtracting the profit from the balance.

Finally, the function returns the profit, principal, and balance.

Task - 3 :

Write a procedure to calculate all accounts’ profit (i.e. profit will be calculated if it satisfies conditions). Use the cursor for loop for this problem. The procedure will insert the appropriate record in its Amounts table.

```
CREATE OR REPLACE PROCEDURE calculate_profit AS
    CURSOR account_cursor IS
        SELECT a.ID, a.OPENINGDATE, p.PROFITRATE, p.GRACEPERIOD
        FROM ACCOUNT a
        JOIN ACCOUNTPROPERTY p
        ON a.ACCCODE = p.ID;

    v_accno ACCOUNT.ID%TYPE;
    v_openingdate ACCOUNT.OPENINGDATE%TYPE;
    v_profitrate ACCOUNTPROPERTY.PROFITRATE%TYPE;
    v_graceperiod ACCOUNTPROPERTY.GRACEPERIOD%TYPE;
    v_duration NUMBER;
    v_profit NUMERIC(10,2);
    v_principal NUMERIC(10,2);
    v_balance NUMERIC(10,2);

BEGIN
    FOR account_rec IN account_cursor
    LOOP
        v_accno := account_rec.ID;
        v_openingdate := account_rec.OPENINGDATE;
        v_profitrate := account_rec.PROFITRATE;
        v_graceperiod := account_rec.GRACEPERIOD;
        v_duration := MONTHS_BETWEEN(SYSDATE, v_openingdate);
    IF v_duration >= v_graceperiod THEN
      v_profit := (SELECT SUM(AMOUNT) FROM TRANSACTION WHERE ACCNO =
v_accno) * (v_profitrate/100) * (v_duration/12);
      v_principal := (SELECT PRINCIPALAMOUNT FROM BALANCE WHERE ACCNO
= v_accno);
      v_balance := v_principal + v_profit;

      INSERT INTO AMOUNTS (ACCNO, PRINCIPALAMOUNT, PROFITAMOUNT,
BALANCE)
      VALUES (v_accno, v_principal, v_profit, v_balance);
    END IF;
  END LOOP;
END;
/
```

Description :

This procedure calculates the profit of each account in the ACCOUNT table.
The procedure starts with a cursor (account_cursor) that selects the ID, opening date, profit rate, and grace period of each account.
In the FOR loop, the procedure iterates through each record returned by the cursor. For each iteration, it calculates the duration (in months) between the current date and the opening date of the account. If the duration is greater than or equal to the grace period, the procedure calculates the profit, principal, and balance of the account. The profit is calculated as the sum of all amounts in the TRANSACTION table for the account multiplied by the profit rate (as a decimal) multiplied by the duration in years. The principal is calculated as the amount in the BALANCE table for the account. The balance is calculated as the sum of the principal and the profit.
Finally, the procedure inserts the values for the account's principal, profit, and balance into the AMOUNTS table.