CSE 4508 – RDBMS Programming Lab

Lab 8

Group 2A

Instructor: Shahriar Ivan

Materials: Prof. Dr. Abu Raihan Mostafa Kamal

A. You have an employee (ID, Name, Salary and Designation) table where salary is an attribute. Try to increase it by 10% for employees having designation “manager” with salary <30000 and decrease it by 10% for “assistant manager” with salary>20000 and show how many rows got affected using an implicit cursor.

B. Create a table **transactions** (User_ID, Amount, T_Date) which stores all bank transactions of all the users in our hypothetical bank. Fill up the table with a few transactions of your choice. Create another table **loan_type** (Scheme, Installment_Number, Charge, Min_Trans). Loan_type will have the loan schemes as shown below. For simplicity, you can store the Scheme as a number, such as 1, 2, or 3 instead of “S-A/S-B/S-C”. Insert only **those 3 specific rows** into the table. Now, create a function that takes as input a User_ID, calculates his/her total transactions, and checks against the loan_type table (use a cursor here) to determine the correct present loan scheme for this person. Return and display the loan_scheme number.

Scheme	No. of Installment	Service Charge for remaining loan	Eligibility
S-A	30	5%	Total Transaction in the last 12 months$\geq$ 2000000
S-B	20	10%	Total Transaction in the last 12 months$\geq$ 1000000
S-C	15	15%	Total Transaction in the last 12 months$\geq$ 500000

Task 1

TABLE CREATION

```
create table Employees(
        id int,
        name varchar(20),
        salary int,
        designation varchar(20),
        constraint emp_pk primary key(id)
);
insert into Employees values(1,'ash',50000,'manager');
insert into Employees values(2,'asha',35000,'assistant-manager');
insert into Employees values(3,'ashar',45000,'assistant-manager');
insert into Employees values(4,'ashari',35000,'assistant-manager');
insert into Employees values(5,'asharin',75000,'manager');
insert into Employees values(6,'asharina',25000,'assistant-manager');
insert into Employees values(7,'ash',5000,'manager');
insert into Employees values(8,'ash',5000,'manager');
```

MAIN FUNCTION

```
declare total_rows number;
begin
update employees
set salary = salary + salary * 0.10
where(
        salary > 30000
        and designation = 'manager'
);
```

```
total_rows := sql % rowcount;

update employees
set salary = salary - salary * 0.10
where(
        salary > 20000 and designation = 'assistant-manager'
);
total_rows := total_rows + sql % rowcount;

if total_rows = 0 then
        dbms_output.put_line('No Employees Selected');
else
        dbms_output.put_line(total_rows || ' Employees Selected');
end if;

end;

\
```

Task 2

TABLE CREATION

```
create table loan_type(
        scheme number check(scheme > 0 and scheme < 4),
        number_installments number,
        charge number,
        total_transactions number,
        constraint loan_pk primary key(scheme)
);
insert into loan_type values(1,30,0.05,2000000);
insert into loan_type values(2,20,0.1,1000000);
insert into loan_type values(3,15,0.15,500000);
create table transaction(
        user_id int,
        amount number,
        t_date date
);
insert into transaction values(1,10000000,DATE '2022-5-10');
insert into transaction values(1,10000000,DATE '2022-6-10');
insert into transaction values(1,5000000,DATE '2022-9-10');
insert into transaction values(2,5000000,DATE '2022-6-10');
insert into transaction values(2,5000000,DATE '2022-5-10');
insert into transaction values(3,10000,DATE '2022-6-10');
insert into transaction values(3,1000,DATE '2022-5-10');
insert into transaction values(3,1000,DATE '2022-6-10');
```

PL/SQL CODE

```
create or replace function scheme_number(uid in number)
return number
is
        scheme_no number;
        total_transactions number;
        min_transactions number;
        flag number:=0;
cursor c_loan_type is
        select scheme,total_transactions from loan_type;
cursor c_transaction is
        select sum(amount) from transaction
where user_id = uid and  (sysdate-t_date < 365);
begin
        open c_transaction;
        fetch c_transaction into total_transactions;
        open c_loan_type;
loop
        fetch c_loan_type into scheme_no,min_transactions;
        exit when c_loan_type%notfound;
if(total_transactions >= min_transactions) then
        flag := 1;
        dbms_output.put_line('scheme no '||scheme_no);
        exit;
end if;
end loop;
```

```
if flag=0 then
        dbms_output.put_line('no scheme');
end if;
close c_loan_type;
close c_transaction;
return scheme_no;
end;
/
```

MAIN FUNCTION

```
declare
        ans number;
        u_id number:='&u_id';
begin
        ans:=scheme_number(u_id);
end;
/
```

—XXX—