**Check past due amount:**

checkDue(dueAmt out number, lateFee out number)

*Algorithm:* The procedure will verify each account to check if the balance if positive and current date has crossed due date from **‘monthly\_bill**’ table; then **5% of charges is added to the account balance**.

Message will be inserted into the **‘message’** table to state that **‘Account is past due’**

It displays the account balance after charges is applied for customer reference.

*Input:* NA

*Output:* Message will be displayed stating that Account will be in due.

CREATE OR REPLACE PROCEDURE checkDue IS

CURSOR chk\_due IS SELECT a.aid,a.account\_balance,m.bid,m.due\_date FROM monthly\_bill m,account a WHERE m.due\_date < sysdate AND a.account\_balance > 0 AND m.aid = a.aid AND m.due\_date IN (SELECT max(m.due\_date) FROM monthly\_bill m,account a WHERE m.due\_date < sysdate AND a.account\_balance > 0 AND m.aid = a.aid

group by m.aid) ;

late\_fee account.account\_balance%type;

v\_aid account.aid%type;

v\_account\_balance account.account\_balance%type;

v\_bid monthly\_bill.bid%type;

v\_due\_date monthly\_bill.due\_date%type;

BEGIN

OPEN chk\_due;

LOOP

FETCH chk\_due INTO

v\_aid, v\_account\_balance,v\_bid,v\_due\_date;

EXIT WHEN chk\_due%notfound;

/\*5% late charge is added to account balance.\*/

late\_fee := (0.05 \* v\_account\_balance);

UPDATE account SET account\_balance = account\_balance + late\_fee WHERE aid = v\_aid;

dbms\_output.put\_line('Account id '||v\_aid||'has been charges with late fee.');

/\*Message table is updated with a new entry.\*/

INSERT INTO message VALUES(message\_id\_seq.nextVal,v\_aid,'Account is past due.',systimestamp);

commit;

END LOOP;

Close Chk\_Due;

EXCEPTION

when no\_data\_found then dbms\_output.put\_line('Data for the speficied account doesnot exists');

when too\_many\_rows then dbms\_output.put\_line('Too many rows');

when others then dbms\_output.put\_line('Error');

END;