

**PROGRAM TITLE:**Create a program in PL/SQL to withdraw money from a bank account. The minimum balance in an account must be 1000. If the account is not used for more than 6 months then deactivate it.

**PROGRAM CODE:**

Table Creation:

**ba:**

```
create table ba(aid number(4),dop date,dol date,balance
number(8),primary key aid);
```

**bastatus:**

```
create table bastatus (aid number(4),status varchar(1),primary
key(aid));
```

Final Tables Created:

**ba:**

	AID	DOP	DOL	BALANCE
-----	-----	-----	-----	-----
	2651	01-01-14	01-01-14	6000
	2561	02-01-15	02-01-15	6000
	1	05-11-15	05-11-15	1200
	2	05-10-15	04-11-16	1250

**bastatus:**

no rows selected

(The Table is empty in the beginning.)

Table Insertion(example):

```
SQL> insert into ba values(&aid, to_date('&dop','dd/MM/yyyy'),
to_date('&dol','dd/MM/yyyy'), &balance);
Enter value for aid: 2
Enter value for dop: 5/10/15
Enter value for dol: 4/11/16
Enter value for balance: 1250
old 1: insert into ba values(&aid, to_date('&dop','dd/MM/yyyy'),
to_date('&dol','dd/MM/yyyy'), &balance)
new 1: insert into ba values(2, to_date('5/10/15','dd/MM/yyyy'),
to_date('4/11/16','dd/MM/yyyy'), 1250)
```

1 row created.

PL/SQL Code:

declare

```
ano number(4);
amt number(8);
bal number(8);
ans varchar2(10);
```

```

yes varchar2(10):='yes';
--cant use these variables anywhere else outside the script
begin

ano:=&ano;
select 'yes' into ans from ba where aid=ano and (sysdate-dol)>182;

if ans = yes then
    insert into bastatus values (ano,'I');
    dbms_output.put_line(ano || 'is deactivated');
end if;

exception
when NO_DATA_FOUND then
    select balance into bal from ba where aid=ano;
    amt:=&amt;
    if bal-amt<1000 then
        dbms_output.put_line('Insufficient Balance');
        --print line for pl/sql
    else
        update ba set balance=bal-amt,dol=(select sysdate from
dual) where aid=ano;
    end if;
end;
/

```

#### OUTPUT:

```

SQL> @bankupdate;
Enter value for ano: 2
old 11: ano:=&ano;
new 11: ano:=2;
Enter value for amt: 200
old 22: amt:=&amt;
new 22: amt:=200;

```

PL/SQL procedure successfully completed.

```
SQL> select * from ba;
```

AID	DOP	DOL	BALANCE
2651	01-01-14	01-01-14	6000
2561	02-01-15	02-01-15	6000
1	05-11-15	05-11-15	1200
2	05-10-15	16-11-16	1050

```

SQL> @bankupdate;
Enter value for ano: 2561
old 11: ano:=&ano;
new 11: ano:=2561;
Enter value for amt: 2000
old 22: amt:=&amt;
new 22: amt:=2000;
2561is deactivated

```

PL/SQL procedure successfully completed.

```
SQL> select * from bastatus;
```

AID	S
2561	I