

6. Create a table **BANK** with the columns ACNO, ACT_NAME, ACT_TYPE and BAL. Specify the Primary Key. Initial BAL must be greater than 500. Write a PL/SQL program to perform debit operation by providing acct_no and amount required. The amount must be greater than 100 and less than 20000 for one transaction. If the account exists and BAL-amount>100 **BANK** table must be updated, otherwise “NO SUFFICIENT BALANCE” message should be displayed. If account number is not present then display “NO SUCH ACCOUNT” message to the user.

6.1 Creating the table and describing its structure

```
CREATE TABLE BANK
(
  ACNO INTEGER PRIMARY KEY,
  ACT_NAME VARCHAR(20) NOT NULL,
  ACT_TYPE VARCHAR(10) NOT NULL,
  BAL NUMBER(10, 2) NOT NULL,
  CHECK (BAL > 500)
);
```

Output

Table created.

Description of BANK table.

```
DESCRIBE BANK;          or          DESC BANK;
```

Name	Null?	Type
ACNO	NOT NULL	NUMBER(38)
ACT_NAME	NOT NULL	VARCHAR2(20)
ACT_TYPE	NOT NULL	VARCHAR2(10)
BAL	NOT NULL	NUMBER(10,2)

6.2 Loading data into the table

```
INSERT INTO BANK VALUES(1001, 'ASHRAY', 'SAVINGS', 15000);
INSERT INTO BANK VALUES(1002, 'NAYANA', 'SAVINGS', 11300);
INSERT INTO BANK VALUES(1003, 'TARUN', 'CURRENT', 50000);
INSERT INTO BANK VALUES(1004, 'ANANYA', 'CURRENT', 34280);
INSERT INTO BANK VALUES(1005, 'BHUVAN', 'SAVINGS', 5000);
INSERT INTO BANK VALUES(1006, 'ISHA', 'CURRENT', 45000);
INSERT INTO BANK VALUES(1007, 'VARSHITH', 'SAVINGS', 9750);
INSERT INTO BANK VALUES(1008, 'RISHAN', 'SAVINGS', 10000);
INSERT INTO BANK VALUES(1009, 'ADVAITH', 'SAVINGS', 15100);
INSERT INTO BANK VALUES(1010, 'SHARAN', 'CURRENT', 12177);
```

Output

10 rows created.

Displaying the values inserted into the BANK table.

```
SELECT *
FROM BANK;
```

ACNO	ACT_NAME	ACT_TYPE	BAL
1001	ASHRAY	SAVINGS	15000
1002	NAYANA	SAVINGS	11300
1003	TARUN	CURRENT	50000
1004	ANANYA	CURRENT	34280
1005	BHUVAN	SAVINGS	5000
1006	ISHA	CURRENT	45000
1007	VARSHITH	SAVINGS	9750
1008	RISHAN	SAVINGS	10000
1009	ADVAITH	SAVINGS	15100
1010	SHARAN	CURRENT	12177

10 rows selected.

6.3 PL/SQL Program

```
SET SERVEROUTPUT ON;
```

DECLARE

```
balance BANK.BAL%type;
acct_no BANK.ACNO%type;
withdraw_amt number(10,2);
```

BEGIN

```
acct_no:=&acct_no;
withdraw_amt:=&withdraw_amt;
select BAL into balance
from BANK
where ACNO=acct_no;

if(balance < withdraw_amt and withdraw_amt>100 and withdraw_amt<20000) then
    dbms_output.put_line('No Sufficient Balance! ' || chr(10) || 'Avaialable Balance = ' || balance);

elsif(withdraw_amt>100 and withdraw_amt<20000) then
    balance:=balance-withdraw_amt;
    dbms_output.put_line('Amount withdrawn! ' || chr(10) || 'New Balance = ' || balance);
    update BANK set BAL=balance where ACNO=acct_no;

elsif(withdraw_amt<100 or withdraw_amt>20000) then
    dbms_output.put_line('Withdraw amount is out of transaction limits!');

end if;
```

EXCEPTION

```
when no_data_found then
dbms_output.put_line('No Such Account..!');
```

```
END;
```

```
/
```

Output**Case 1: Successful debit transaction.**

```
SQL>@"BANK.SQL"
```

```
Enter value for acct_no: 1001
```

```
old 6: acct_no:=&acct_no;
```

```
new 6: acct_no:=1001;
```

```
Enter value for withdraw_amt: 5000
```

```
old 7: withdraw_amt:=&withdraw_amt;
```

```
new 7: withdraw_amt:=5000;
```

```
Amount withdrawn!
```

```
New Balance = 10000
```

```
PL/SQL procedure successfully completed.
```

```
Displaying the updated records of the BANK table.
```

```
SELECT *
```

```
FROM BANK;
```

ACNO	ACT_NAME	ACT_TYPE	BAL
1001	ASHRAY	SAVINGS	10000
1002	NAYANA	SAVINGS	11300
1003	TARUN	CURRENT	50000
1004	ANANYA	CURRENT	34280
1005	BHUVAN	SAVINGS	5000
1006	ISHA	CURRENT	45000
1007	VARSHITH	SAVINGS	9750
1008	RISHAN	SAVINGS	10000
1009	ADVAITH	SAVINGS	15100
1010	SHARAN	CURRENT	12177

```
10 rows selected.
```

Case 2: When the account number is not present in the BANK table.

```
SQL>@"BANK.SQL"
```

```
Enter value for acct_no: 1011
```

```
old 6: acct_no:=&acct_no;
```

```
new 6: acct_no:=1011;
```

```
Enter value for withdraw_amt: 1000
```

```
old 7: withdraw_amt:=&withdraw_amt;
```

```
new 7: withdraw_amt:=1000;
```

```
No Such Account..!
```

```
PL/SQL procedure successfully completed.
```

Case 3 a: When amount to be withdrawn is less than 100.

```
SQL>@"BANK.SQL"
```

```
Enter value for acct_no: 1001
```

```
old 7: acct_no:=&acct_no;
```

```
new 7: acct_no:=1001;
```

```
Enter value for withdraw_amt: 50
```

```
old 8: withdraw_amt:=&withdraw_amt;
new 8: withdraw_amt:=50;

Withdraw amount is out of transaction limits!

PL/SQL procedure successfully completed.
```

Case 3 b: When amount to be withdrawn is more than 20000.

```
SQL> @"BANK.SQL"

Enter value for acct_no: 1001
old 7: acct_no:=&acct_no;
new 7: acct_no:=1001;

Enter value for withdraw_amt: 22000
old 8: withdraw_amt:=&withdraw_amt;
new 8: withdraw_amt:=22000;

Withdraw amount is out of transaction limits!

PL/SQL procedure successfully completed.
```

Case 4: When the account does not have sufficient balance.

```
Enter value for acct_no: 1001
old 7: acct_no:=&acct_no;
new 7: acct_no:=1001;

Enter value for withdraw_amt: 15000
old 8: withdraw_amt:=&withdraw_amt;
new 8: withdraw_amt:=15000;

No Sufficient Balance!

Available Balance = 10000
```

Date of Submission	Remarks
Signature of the Lecturer	