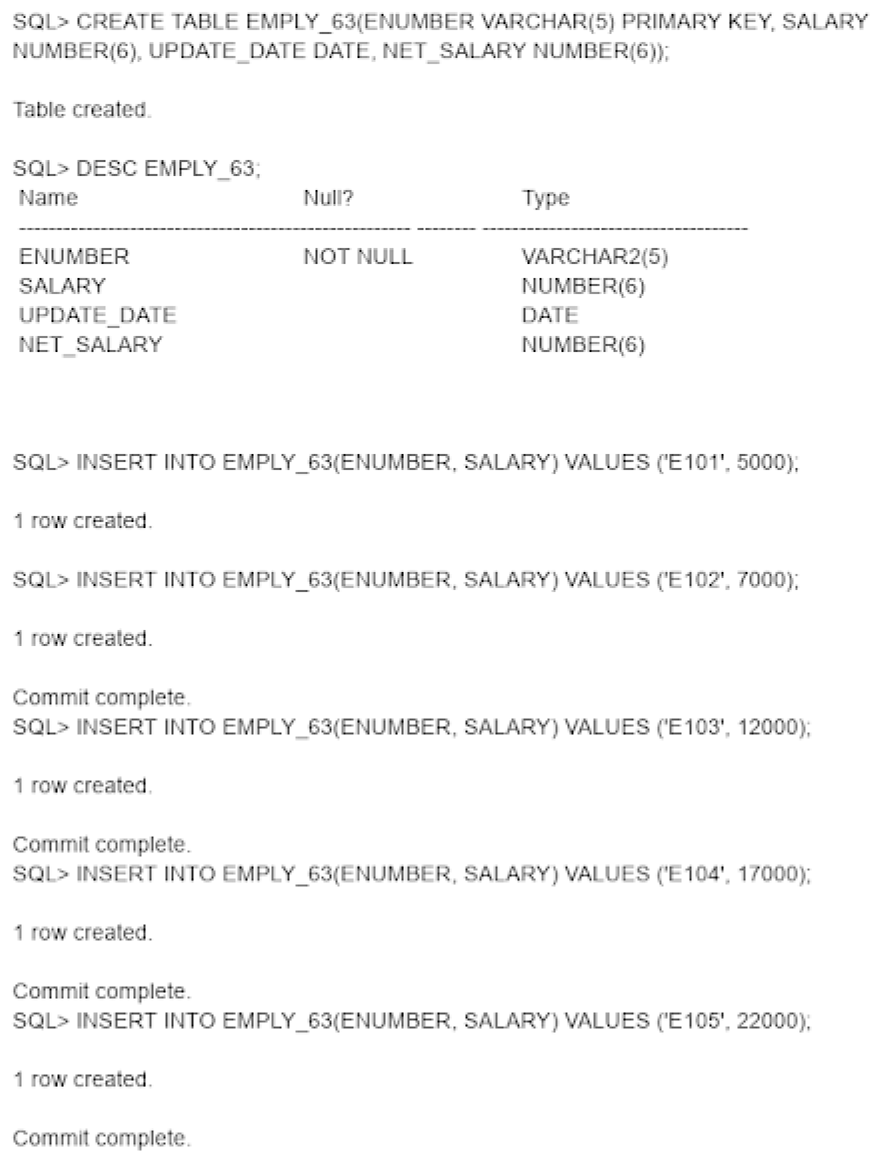
```
SQL> CREATE TABLE CUSTOMER_23 (NAME VARCHAR(10) PRIMARY KEY, ADDRESS
VARCHAR(15));

Table created.

SQL> DESC CUSTOMER_23;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 NAME                                      NOT NULL VARCHAR2(10)
 ADDRESS                                            VARCHAR2(15)



SQL> CREATE TABLE BRANCH_23 (BRANCH_ID VARCHAR(5) PRIMARY KEY,
BRANCH_CITY VARCHAR(10));

Table created.

SQL> DESC BRANCH_23;
```

Name Null? Type



```
SQL> CREATE TABLE EMPLY_63(ENUMBER VARCHAR(5) PRIMARY KEY, SALARY
NUMBER(6), UPDATE_DATE DATE, NET_SALARY NUMBER(6));

Table created.

SQL> DESC EMPLY_63;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ENUMBER                                   NOT NULL VARCHAR2(5)
 SALARY                                             NUMBER(6)
 UPDATE_DATE                                        DATE
 NET_SALARY                                         NUMBER(6)


SQL> INSERT INTO EMPLY_63(ENUMBER, SALARY) VALUES ('E101', 5000);

1 row created.

SQL> INSERT INTO EMPLY_63(ENUMBER, SALARY) VALUES ('E102', 7000);

1 row created.

Commit complete.
SQL> INSERT INTO EMPLY_63(ENUMBER, SALARY) VALUES ('E103', 12000);

1 row created.

Commit complete.
SQL> INSERT INTO EMPLY_63(ENUMBER, SALARY) VALUES ('E104', 17000);

1 row created.

Commit complete.
SQL> INSERT INTO EMPLY_63(ENUMBER, SALARY) VALUES ('E105', 22000);

1 row created.

Commit complete.
```

Open

```
 --------------------------------------- -------- ----------------------------
 BRANCH_ID                          NOT NULL      VARCHAR2(5)
 BRANCH_CITY                                       VARCHAR2(10)



SQL> CREATE TABLE ACCOUNT_23 (NAME VARCHAR(10) REFERENCES
CUSTOMER_23(NAME), ACC_NO NUMBER(10) PRIMARY KEY, BALANCE NUMBER(6)
CHECK(BALANCE>=500), BRANCH_ID VARCHAR(5) REFERENCES
BRANCH_23(BRANCH_ID));

Table created.

SQL> DESC ACCOUNT_23;
 Name                               Null?         Type
 --------------------------------------- -------- ----------------------------
 NAME                                              VARCHAR2(10)
 ACC_NO                NOT NULL                   NUMBER(10)
 BALANCE                                          NUMBER(6)
 BRANCH_ID                                         VARCHAR2(5)



SQL> INSERT INTO CUSTOMER_23 VALUES('&NAME','&ADDRESS');
Enter value for name: ANIL
Enter value for address: TVM
old   1: INSERT INTO CUSTOMER_23 VALUES('&NAME','&ADDRESS')
new   1: INSERT INTO CUSTOMER_23 VALUES('ANIL','TVM')

1 row created.


SQL> SELECT * FROM CUSTOMER_23;

NAME            ADDRESS
---------- ---------------
ANIL    TVM
BHEEM           KOLLAM
BABU            WAYANAD
RAMESH          ALUVA
KUMAR           THRISSUR
```

```
SQL> INSERT INTO BRANCH_23 VALUES('&ID','&CITY');
Enter value for id: BR123
Enter value for city: ANGAMALY
old   1: INSERT INTO BRANCH_23 VALUES('&ID','&CITY')
new   1: INSERT INTO BRANCH_23 VALUES('BR123','ANGAMALY')

1 row created.


SQL> SELECT * FROM BRANCH_23;

BRANC        BRANCH_CIT
----- ----------------------------------
BR123        ANGAMALY
BR234        PUNALOOR
BR345        MANNUTHY
BR432        PUNALOOR
BR785        ANGAMALY
BR429        ANGAMALY

6 rows selected.
```

bank.sql

```
DECLARE
CNT NUMBER(2);
N CUSTOMER_23.NAME%TYPE;
A CUSTOMER_23.ADDRESS%TYPE;
BR ACCOUNT_23.BRANCH_ID%TYPE;
B ACCOUNT_23.BALANCE%TYPE;
ACCNO ACCOUNT_23.ACC_NO%TYPE;
BEGIN
        N := '&NAME';
        A:= '&ADDRESS';
        ACCNO := SEQ.NEXTVAL;
        B := &BALANCE;
        BR := '&BRANCH_ID';

        SELECT COUNT(*) INTO CNT FROM CUSTOMER_23 WHERE NAME=N;

        IF (CNT = 0) THEN

            INSERT INTO CUSTOMER_23 VALUES(N,A);
        END IF;

        INSERT INTO ACCOUNT_23 VALUES(N,ACCNO,B,BR);
END;
```

banktrig.sql

```
CREATE OR REPLACE TRIGGER BANKTRIGGER
BEFORE INSERT ON ACCOUNT_23
FOR EACH ROW
DECLARE
CNT NUMBER(2);
BEGIN
        SELECT COUNT(*) INTO CNT FROM ACCOUNT_23 WHERE NAME=:NEW.NAME;
        IF CNT>=3 THEN
          RAISE_APPLICATION_ERROR(-20001,' ALREADY HAS 3 ACCOUNTS');
        END IF;
END;
```

OUTPUT

```
SQL> CREATE SEQUENCE SEQ START WITH 1001 INCREMENT BY 1;

Sequence created.

SQL> @banktrig.sql;
 13  /

Trigger created.



SQL> @bank.sql
 24  /
Enter value for name: JASON
old   9:            N := '&NAME';
new   9:            N := 'JASON';
Enter value for address: THRISSUR
old  10:            A:= '&ADDRESS';
new  10:            A:= 'THRISSUR';
```

```
Enter value for balance: 12000
old  12:         B := &BALANCE;
new  12:         B := 12000;
Enter value for branch_id: BR345
old  13:         BR := '&BRANCH_ID';
new  13:         BR := 'BR345';

PL/SQL procedure successfully completed.

Commit complete.


SQL> SELECT * FROM ACCOUNT_23;

NAME           ACC_NO     BALANCE        BRANC
---------- ---------- ---------- -------------------------------------
JASON           1004           12000           BR345

SQL> SELECT * FROM CUSTOMER_23;

NAME           ADDRESS
---------- ---------------
ANIL    TVM
BHEEM          KOLLAM
BABU           WAYANAD
RAMESH         ALUVA
KUMAR          THRISSUR
JASON          THRISSUR

6 rows selected.



SQL> @bank.sql
 24  /
Enter value for name: ANIL
old   9:         N := '&NAME';
new   9:         N := 'ANIL';
Enter value for address: TVM
old  10:         A:= '&ADDRESS';
new  10:         A:= 'TVM';
Enter value for balance: 25000
old  12:         B := &BALANCE;
new  12:         B := 25000;
```

```
Enter value for branch_id: BR234
old  13:          BR := '&BRANCH_ID';
new  13:          BR := 'BR234';

PL/SQL procedure successfully completed.

Commit complete.


SQL> SELECT * FROM ACCOUNT_23;

NAME           ACC_NO     BALANCE BRANC
---------- ---------- ---------- -----
JASON            1001            12000 BR345
ANIL      1002           25000 BR234
ANIL      1003           45000 BR123
ANIL      1004           45000 BR123

SQL> @bank.sql
 25  /
Enter value for name: ANIL
old   9:          N := '&NAME';
new   9:          N := 'ANIL';
Enter value for address: TVM
old  10:          A:= '&ADDRESS';
new  10:          A:= 'TVM';
Enter value for balance: 33000
old  12:          B := &BALANCE;
new  12:          B := 33000;
Enter value for branch_id: BR785
old  13:          BR := '&BRANCH_ID';
new  13:          BR := 'BR785';
DECLARE
*
ERROR at line 1:
ORA-20001:  ALREADY HAS 3 ACCOUNTS
ORA-06512: at "CS27021.BANKTRIGGER", line 6
ORA-04088: error during execution of trigger 'CS27021.BANKTRIGGER'
ORA-06512: at line 22
```

```
SQL> SELECT BRANCH_ID,COUNT(BRANCH_ID) FROM ACCOUNT_23 GROUP BY
BRANCH_ID HAVING COUNT(BRANCH_ID)=(SELECT MAX(COUNT(BRANCH_ID)) FROM
ACCOUNT_23 GROUP BY BRANCH_ID);

BRANC           COUNT(BRANCH_ID)
----- -----------------------------------------
BR123                        2


SQL> SELECT BRANCH_CITY, COUNT(BRANCH_CITY) FROM BRANCH_23 GROUP BY
BRANCH_CITY HAVING COUNT(BRANCH_CITY)=(SELECT MAX(COUNT(BRANCH_CITY))
FROM BRANCH_23 GROUP BY BRANCH_CITY);

BRANCH_CIT COUNT(BRANCH_CITY)
---------- ------------------
ANGAMALY                  3
```