```
SQL> @D:\01-College\databaselab\ex1.sql
SQL> DROP TABLE Reservations;

Table dropped.

SQL> DROP TABLE Boat;

Table dropped.

SQL> DROP TABLE Sailor;

Table dropped.

SQL> DROP TABLE Tourist;

Table dropped.

SQL>
SQL> REM ************************************
SQL> REM Creating a table Boat using the given attributes
SQL>
SQL> CREATE TABLE Boat(
  2        BID varchar2(4) CONSTRAINT bid_pk PRIMARY KEY CONSTRAINT
bid_start CHECK(BID LIKE 'B%'),
  3        BNAME varchar2(15),
  4        TYPE varchar2(4) CONSTRAINT b_tchk CHECK(TYPE
IN('LUX','CAR','CRU')),
  5        MAX_CAP number(2),
  6        PRICE number(3),
  7        COLOR varchar2(10)
  8  )
  9  /

Table created.

SQL>
SQL> REM Displaying Boat
SQL>
SQL> desc Boat
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 BID                                       NOT NULL VARCHAR2(4)
 BNAME                                              VARCHAR2(15)
 TYPE                                               VARCHAR2(4)
 MAX_CAP                                            NUMBER(2)
 PRICE                                              NUMBER(3)
 COLOR                                              VARCHAR2(10)

SQL>
SQL> REM inserting values into boats table
SQL>
SQL> INSERT INTO Boat VALUES('B100','speedboat','LUX',90,23,'yellow');

1 row created.

SQL> INSERT INTO Boat VALUES('B101','sail','LUX',90,23,'yellow');

1 row created.
```

```
SQL> INSERT INTO Boat VALUES('B102','oceana','CAR',90,23,'yellow');

1 row created.

SQL> INSERT INTO Boat VALUES('B103','rivera','CRU',90,23,'yellow');

1 row created.

SQL> INSERT INTO Boat VALUES('B104','sailboat','CAR',90,23,'blue');

1 row created.

SQL> INSERT INTO Boat VALUES('B105','yatch','REM',90,23,'red');
INSERT INTO Boat VALUES('B105','yatch','REM',90,23,'red')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.B_TCHK) violated


SQL> INSERT INTO Boat VALUES('B106','yatch','LUX',90,23,'yellow');

1 row created.

SQL> INSERT INTO Boat VALUES('B100','sail','LUX',90,23,'yellow');
INSERT INTO Boat VALUES('B100','sail','LUX',90,23,'yellow')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.BID_PK) violated


SQL> INSERT INTO Boat VALUES('100','sail','LUX',90,23,'yellow');
INSERT INTO Boat VALUES('100','sail','LUX',90,23,'yellow')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.BID_START) violated


SQL>
SQL> SELECT * FROM Boat;

BID  BNAME           TYPE    MAX_CAP      PRICE COLOR
---- --------------- ---- ---------- ---------- ----------
B100 speedboat       LUX          90         23 yellow
B101 sail            LUX          90         23 yellow
B102 oceana          CAR          90         23 yellow
B103 rivera          CRU          90         23 yellow
B104 sailboat        CAR          90         23 blue
B106 yatch           LUX          90         23 yellow

6 rows selected.

SQL>
SQL> REM ************************************
SQL> REM Creating a table Sailor using the given attributes
SQL>
SQL> CREATE TABLE Sailor(
  2        SID varchar2(4) CONSTRAINT sid_pk PRIMARY KEY,CONSTRAINT
sid_start CHECK(SID LIKE 'S%'),
  3        SNAME varchar2(15),
```

```
  4        RATING char(1) CONSTRAINT rc_chk CHECK(RATING
IN('A','B','C')),
  5        DOB date
  6  )
  7  /

Table created.

SQL>
SQL> REM Displaying Sailor
SQL>
SQL> desc Sailor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 SID                                       NOT NULL VARCHAR2(4)
 SNAME                                              VARCHAR2(15)
 RATING                                             CHAR(1)
 DOB                                                DATE

SQL>
SQL> REM inserting values into sailor table
SQL>
SQL> INSERT INTO Sailor VALUES('S100','Sam','A','11-jan-90');

1 row created.

SQL> INSERT INTO Sailor VALUES('S101','Som','B','15-feb-80');

1 row created.

SQL> INSERT INTO Sailor VALUES('S102','Sam','F','14-feb-85');
INSERT INTO Sailor VALUES('S102','Sam','F','14-feb-85')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.RC_CHK) violated


SQL> INSERT INTO Sailor VALUES('S103','Sai','C','4-oct-95');

1 row created.

SQL> INSERT INTO Sailor VALUES('S104','Shree','C','9-jul-83');

1 row created.

SQL> INSERT INTO Sailor VALUES('S100','Sai','C','13-jun-94');
INSERT INTO Sailor VALUES('S100','Sai','C','13-jun-94')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SID_PK) violated


SQL> INSERT INTO Sailor VALUES('100','Sri','C','12-aug-93');
INSERT INTO Sailor VALUES('100','Sri','C','12-aug-93')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.SID_START) violated
```

```
SQL> INSERT INTO Sailor VALUES('S105','Shree','C','5-nov-88');

1 row created.

SQL> INSERT INTO Sailor VALUES('S106','Shree','C','18-dec-90');

1 row created.

SQL>
SQL> SELECT * FROM Sailor;

SID  SNAME           R DOB
---- --------------- - ---------
S100 Sam             A 11-JAN-90
S101 Som             B 15-FEB-80
S103 Sai             C 04-OCT-95
S104 Shree           C 09-JUL-83
S105 Shree           C 05-NOV-88
S106 Shree           C 18-DEC-90

6 rows selected.

SQL>
SQL>
SQL> REM ************************************
SQL> REM Creating a table Tourist using the given attributes
SQL>
SQL> CREATE TABLE Tourist(
  2        TID varchar2(4) CONSTRAINT t_id PRIMARY KEY,CONSTRAINT
tid_start CHECK(TID LIKE 'T%'),
  3        TNAME varchar2(10),
  4        ADDR varchar2(20),
  5        PHN number(10),
  6        DOB date
  7  )
  8  /

Table created.

SQL>
SQL> REM Displaying Tourist
SQL>
SQL> desc Tourist
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 TID                                       NOT NULL VARCHAR2(4)
 TNAME                                              VARCHAR2(10)
 ADDR                                               VARCHAR2(20)
 PHN                                                NUMBER(10)
 DOB                                                DATE

SQL>
SQL>
SQL> REM inserting values into tourist table
SQL>
SQL> INSERT INTO Tourist VALUES('T100','Shreya','Chennai',1982828773,'11-
feb-90');

1 row created.
```

```
SQL> INSERT INTO Tourist VALUES('T101','Sri','Saidapet',1289213899,'19-
jan-89');

1 row created.

SQL> INSERT INTO Tourist VALUES('T102','Sam','Adayar',1199288822,'1-oct-
93');

1 row created.

SQL> INSERT INTO Tourist VALUES('T103','Sri','Saidapet',1289213899,'14-
aug-99');

1 row created.

SQL> INSERT INTO Tourist VALUES('T104','Sri','Saidapet',1289213899,'15-
aug-99');

1 row created.

SQL> INSERT INTO Tourist VALUES('T105','Sri','Saidapet',1289213899,'18-
aug-99');

1 row created.

SQL> INSERT INTO Tourist VALUES('T106','Sri','Saidapet',1289213899,'13-
aug-99');

1 row created.

SQL> INSERT INTO Tourist VALUES('100','Sri','Saidapet',1289213899,'12-
aug-99');
INSERT INTO Tourist VALUES('100','Sri','Saidapet',1289213899,'12-aug-99')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.TID_START) violated


SQL> INSERT INTO Tourist VALUES('T100','Sam','Adayar',1199288822,'19-sep-
01');
INSERT INTO Tourist VALUES('T100','Sam','Adayar',1199288822,'19-sep-01')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.T_ID) violated


SQL>
SQL> SELECT * FROM Tourist;

TID  TNAME      ADDR                       PHN DOB
---- ---------- -------------------- ---------- ---------
T100 Shreya     Chennai              1982828773 11-FEB-90
T101 Sri        Saidapet             1289213899 19-JAN-89
T102 Sam        Adayar               1199288822 01-OCT-93
T103 Sri        Saidapet             1289213899 14-AUG-99
T104 Sri        Saidapet             1289213899 15-AUG-99
T105 Sri        Saidapet             1289213899 18-AUG-99
T106 Sri        Saidapet             1289213899 13-AUG-99
```

```
7 rows selected.

SQL>
SQL> REM ************************************
SQL> REM Creating a table Reservations using the given attributes
SQL>
SQL> CREATE TABLE Reservations(
  2        T_ID varchar2(4) CONSTRAINT rt_fk REFERENCES Tourist(TID),
  3        S_ID varchar2(4) CONSTRAINT rs_fk REFERENCES Sailor(SID),
  4        B_ID varchar2(4) CONSTRAINT rb_fk REFERENCES Boat(BID),
  5        NOP number(2),
  6        D_RES date,
  7        D_SAIL date,
  8        CONSTRAINT dt_chk CHECK(D_SAIL-D_RES>12),
  9        CONSTRAINT res_pk PRIMARY KEY(T_ID,S_ID,D_SAIL)
 10  )
 11  /

Table created.

SQL> REM Displaying Reservations
SQL>
SQL> desc Reservations
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 T_ID                                      NOT NULL VARCHAR2(4)
 S_ID                                      NOT NULL VARCHAR2(4)
 B_ID                                               VARCHAR2(4)
 NOP                                                NUMBER(2)
 D_RES                                              DATE
 D_SAIL                                    NOT NULL DATE

SQL>
SQL>
SQL> REM inserting values into reservations table
SQL>
SQL> INSERT INTO Reservations VALUES('T100','S100','B100',19,'11-jan-
20','12-jan-20');
INSERT INTO Reservations VALUES('T100','S100','B100',19,'11-jan-20','12-
jan-20')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.DT_CHK) violated


SQL> INSERT INTO Reservations VALUES('T101','S102','B101',12,'12-jan-
20','15-feb-20');
INSERT INTO Reservations VALUES('T101','S102','B101',12,'12-jan-20','15-
feb-20')
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.RS_FK) violated - parent key not
found


SQL> INSERT INTO Reservations VALUES('T102','S103','B100',18,'13-jan-
20','15-feb-20');

1 row created.
```

```
SQL> INSERT INTO Reservations VALUES('T101','S103','B102',17,'14-jan-
20','15-feb-20');

1 row created.

SQL> INSERT INTO Reservations VALUES('T102','S105','B102',18,'15-jan-
20','15-feb-20');

1 row created.

SQL> INSERT INTO Reservations VALUES('T105','S100','B100',18,'16-jan-
20','15-feb-20');

1 row created.

SQL> INSERT INTO Reservations VALUES('T101','S104','B101',18,'17-jan-
20','15-feb-20');

1 row created.

SQL> INSERT INTO Reservations VALUES('T101','S103','B102',18,'18-jan-
20','15-feb-20');
INSERT INTO Reservations VALUES('T101','S103','B102',18,'18-jan-20','15-
feb-20')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.RES_PK) violated


SQL> INSERT INTO Reservations VALUES('T100','S101','B100',18,'19-jan-
20','15-feb-20');

1 row created.

SQL> INSERT INTO Reservations VALUES('T102','S103','B103',18,'19-jan-
20','15-feb-20');
INSERT INTO Reservations VALUES('T102','S103','B103',18,'19-jan-20','15-
feb-20')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.RES_PK) violated


SQL>
SQL> SELECT * FROM Reservations;

T_ID S_ID B_ID        NOP D_RES     D_SAIL
---- ---- ---- ---------- --------- ---------
T102 S103 B100         18 13-JAN-20 15-FEB-20
T101 S103 B102         17 14-JAN-20 15-FEB-20
T102 S105 B102         18 15-JAN-20 15-FEB-20
T105 S100 B100         18 16-JAN-20 15-FEB-20
T101 S104 B101         18 17-JAN-20 15-FEB-20
T100 S101 B100         18 19-JAN-20 15-FEB-20

6 rows selected.

SQL>
SQL>
```

```
SQL>
SQL> @D:\01-College\databaselab\ex1b.sql
SQL> REM ************************************************
SQL> REM 8. Mentioning number of children
SQL>
SQL> ALTER TABLE Reservations
  2  ADD NO_CHILDREN number(2)
  3  /

Table altered.

SQL>
SQL> desc Reservations;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 T_ID                                      NOT NULL VARCHAR2(4)
 S_ID                                      NOT NULL VARCHAR2(4)
 B_ID                                               VARCHAR2(4)
 NOP                                                NUMBER(2)
 D_RES                                              DATE
 D_SAIL                                    NOT NULL DATE
 NO_CHILDREN                                        NUMBER(2)

SQL>
SQL> REM inserting into reservations table
SQL>
SQL> INSERT INTO Reservations VALUES('T100','S101','B102',19,'11-jan-
20','11-feb-20',3);

1 row created.

SQL> INSERT INTO Reservations VALUES('T101','S101','B102',19,'13-jan-
20','12-feb-20',2);

1 row created.

SQL>
SQL> SELECT * FROM Reservations;

T_ID S_ID B_ID        NOP D_RES     D_SAIL    NO_CHILDREN
---- ---- ---- ---------- --------- --------- -----------
T102 S103 B100         18 13-JAN-20 15-FEB-20
T101 S103 B102         17 14-JAN-20 15-FEB-20
T102 S105 B102         18 15-JAN-20 15-FEB-20
T105 S100 B100         18 16-JAN-20 15-FEB-20
T101 S104 B101         18 17-JAN-20 15-FEB-20
T100 S101 B100         18 19-JAN-20 15-FEB-20
T100 S101 B102         19 11-JAN-20 11-FEB-20           3
T101 S101 B102         19 13-JAN-20 12-FEB-20           2

8 rows selected.

SQL>
SQL> REM ************************************************
SQL> REM 9. to make width of tourist name inadequate
SQL> ALTER TABLE Tourist
  2  MODIFY TNAME varchar(35)
  3  /
```

```
Table altered.

SQL> desc Tourist;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 TID                                       NOT NULL VARCHAR2(4)
 TNAME                                              VARCHAR2(35)
 ADDR                                               VARCHAR2(20)
 PHN                                                NUMBER(10)
 DOB                                                DATE

SQL>
SQL> REM ************************************************
SQL> REM 10. to make reserve date not null
SQL> ALTER TABLE Reservations
  2  MODIFY d_res NOT NULL
  3  /

Table altered.

SQL>
SQL> desc Reservations;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 T_ID                                      NOT NULL VARCHAR2(4)
 S_ID                                      NOT NULL VARCHAR2(4)
 B_ID                                               VARCHAR2(4)
 NOP                                                NUMBER(2)
 D_RES                                     NOT NULL DATE
 D_SAIL                                    NOT NULL DATE
 NO_CHILDREN                                        NUMBER(2)

SQL>
SQL> REM inserting into reservations table
SQL>
SQL> INSERT INTO Reservations VALUES('T101','S103','B100',4,NULL,'21-feb-
20',6);
INSERT INTO Reservations VALUES('T101','S103','B100',4,NULL,'21-feb-
20',6)
                                                    *
ERROR at line 1:
ORA-01400: cannot insert NULL into ("SYSTEM"."RESERVATIONS"."D_RES")


SQL> INSERT INTO Reservations VALUES('T101','S102','B100',4,NULL,'21-feb-
20',1);
INSERT INTO Reservations VALUES('T101','S102','B100',4,NULL,'21-feb-
20',1)
                                                    *
ERROR at line 1:
ORA-01400: cannot insert NULL into ("SYSTEM"."RESERVATIONS"."D_RES")


SQL>
SQL> SELECT * FROM Reservations;

T_ID S_ID B_ID        NOP D_RES     D_SAIL    NO_CHILDREN
---- ---- ---- ---------- --------- --------- -----------
```

```
T102 S103 B100         18 13-JAN-20 15-FEB-20
T101 S103 B102         17 14-JAN-20 15-FEB-20
T102 S105 B102         18 15-JAN-20 15-FEB-20
T105 S100 B100         18 16-JAN-20 15-FEB-20
T101 S104 B101         18 17-JAN-20 15-FEB-20
T100 S101 B100         18 19-JAN-20 15-FEB-20
T100 S101 B102         19 11-JAN-20 11-FEB-20           3
T101 S101 B102         19 13-JAN-20 12-FEB-20           2

8 rows selected.

SQL>
SQL> REM ************************************************
SQL> REM 11. DOB of a tourist can be addressed later
SQL> INSERT INTO Tourist VALUES('T099','Shreya','Chennai',9182772768,'11-
oct-2001');

1 row created.

SQL> ALTER TABLE Tourist
  2  DROP COLUMN DOB
  3  /

Table altered.

SQL>
SQL> desc Tourist;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 TID                                       NOT NULL VARCHAR2(4)
 TNAME                                              VARCHAR2(35)
 ADDR                                               VARCHAR2(20)
 PHN                                                NUMBER(10)

SQL> INSERT INTO Tourist VALUES('T099','Shreya','Chennai',9182772768,'11-
oct-2001');
INSERT INTO Tourist VALUES('T099','Shreya','Chennai',9182772768,'11-oct-
2001')
           *
ERROR at line 1:
ORA-00913: too many values


SQL>
SQL> SELECT * FROM Tourist;

TID  TNAME                               ADDR                      PHN
---- ---------------------------------- -------------------- ----------
T100 Shreya                             Chennai              1982828773
T101 Sri                                Saidapet             1289213899
T102 Sam                                Adayar               1199288822
T103 Sri                                Saidapet             1289213899
T104 Sri                                Saidapet             1289213899
T105 Sri                                Saidapet             1289213899
T106 Sri                                Saidapet             1289213899
T099 Shreya                             Chennai              9182772768

8 rows selected.
```

```
SQL>
SQL>
SQL> REM ***************************************************
SQL> REM 12. rating for sailor D has to be added
SQL>
SQL> ALTER TABLE Sailor
  2  DROP CONSTRAINT rc_chk
  3  /

Table altered.

SQL> ALTER TABLE Sailor
  2  ADD CONSTRAINT rc_chkn CHECK(RATING IN('A','B','C','D'))
  3  /

Table altered.

SQL>
SQL> desc Sailor;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 SID                                       NOT NULL VARCHAR2(4)
 SNAME                                              VARCHAR2(15)
 RATING                                             CHAR(1)
 DOB                                                DATE

SQL> INSERT INTO Sailor VALUES('S199','Sam','D','02-jan-1980');

1 row created.

SQL> INSERT INTO Sailor VALUES('S199','Sam','J','02-jan-1980');
INSERT INTO Sailor VALUES('S199','Sam','J','02-jan-1980')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.RC_CHKN) violated


SQL>
SQL> SELECT * FROM Sailor;

SID  SNAME           R DOB
---- --------------- - ---------
S100 Sam             A 11-JAN-90
S101 Som             B 15-FEB-80
S103 Sai             C 04-OCT-95
S104 Shree           C 09-JUL-83
S105 Shree           C 05-NOV-88
S106 Shree           C 18-DEC-90
S199 Sam             D 02-JAN-80

7 rows selected.

SQL>
SQL> REM ***************************************************
SQL> REM 13. all luxurious boats are yellow
SQL> ALTER TABLE Boat
  2  DROP CONSTRAINT b_tchk
  3  /
```

```
Table altered.

SQL> ALTER TABLE Boat
  2  ADD CONSTRAINT typ_chk CHECK((TYPE IN ('LUX') AND COLOR IN
('yellow')) OR TYPE IN('CAR','CRU'))
  3  /

Table altered.

SQL>
SQL> desc Boat;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-------
 BID                                       NOT NULL VARCHAR2(4)
 BNAME                                              VARCHAR2(15)
 TYPE                                               VARCHAR2(4)
 MAX_CAP                                            NUMBER(2)
 PRICE                                              NUMBER(3)
 COLOR                                              VARCHAR2(10)

SQL> INSERT INTO Boat VALUES('B123','sailboat','LUX',20,6.98,'Blue');
INSERT INTO Boat VALUES('B123','sailboat','LUX',20,6.98,'Blue')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.TYP_CHK) violated


SQL> INSERT INTO Boat VALUES('B123','lakeboat','LUX',20,6.98,'yellow');

1 row created.

SQL>
SQL> SELECT * FROM Boat;

BID  BNAME           TYPE    MAX_CAP      PRICE COLOR
---- --------------- ---- ---------- ---------- ----------
B100 speedboat       LUX          90         23 yellow
B101 sail            LUX          90         23 yellow
B102 oceana          CAR          90         23 yellow
B103 rivera          CRU          90         23 yellow
B104 sailboat        CAR          90         23 blue
B106 yatch           LUX          90         23 yellow
B123 lakeboat        LUX          20          7 yellow

7 rows selected.

SQL>
SQL> REM ********************************************************
SQL> REM 14. Each boat should have a different name
SQL> ALTER TABLE Boat
  2  ADD CONSTRAINT n_uniq UNIQUE(BNAME)
  3  /

Table altered.

SQL>
SQL> desc Boat;
 Name                                      Null?    Type
```

```
 ----------------------------------------- -------- ----------------------
-------
 BID                                       NOT NULL VARCHAR2(4)
 BNAME                                              VARCHAR2(15)
 TYPE                                               VARCHAR2(4)
 MAX_CAP                                            NUMBER(2)
 PRICE                                              NUMBER(3)
 COLOR                                              VARCHAR2(10)

SQL> INSERT INTO Boat VALUES('B123','sailboat','LUX',20,6.98,'Blue');
INSERT INTO Boat VALUES('B123','sailboat','LUX',20,6.98,'Blue')
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.TYP_CHK) violated


SQL> INSERT INTO Boat VALUES('B170','woodboat','CRU',20,8,'Yellow');

1 row created.

SQL>
SQL>
SQL> SELECT * FROM Boat;

BID  BNAME           TYPE    MAX_CAP      PRICE COLOR
---- --------------- ---- ---------- ---------- ----------
B100 speedboat       LUX          90         23 yellow
B101 sail            LUX          90         23 yellow
B102 oceana          CAR          90         23 yellow
B103 rivera          CRU          90         23 yellow
B104 sailboat        CAR          90         23 blue
B106 yatch           LUX          90         23 yellow
B123 lakeboat        LUX          20          7 yellow
B170 woodboat        CRU          20          8 Yellow

8 rows selected.

SQL>
SQL>
SQL> REM ******************************************************
SQL> REM 15. Ensuring that all details of a sailor is deleted on deletion
of boat/sailor details
SQL>
SQL> ALTER TABLE Reservations DROP CONSTRAINT rs_fk;

Table altered.

SQL> ALTER TABLE Reservations DROP CONSTRAINT rb_fk;

Table altered.

SQL> ALTER TABLE Reservations ADD CONSTRAINT rb_fk FOREIGN KEY(B_ID)
REFERENCES Boat(BID) ON DELETE CASCADE;

Table altered.

SQL> ALTER TABLE Reservations ADD CONSTRAINT rs_fk FOREIGN KEY(S_ID)
REFERENCES Sailor(SID) ON DELETE CASCADE;

Table altered.
```

```
SQL>
SQL> SELECT * FROM Reservations;

T_ID S_ID B_ID        NOP D_RES     D_SAIL    NO_CHILDREN
---- ---- ---- ---------- --------- --------- -----------
T102 S103 B100         18 13-JAN-20 15-FEB-20
T101 S103 B102         17 14-JAN-20 15-FEB-20
T102 S105 B102         18 15-JAN-20 15-FEB-20
T105 S100 B100         18 16-JAN-20 15-FEB-20
T101 S104 B101         18 17-JAN-20 15-FEB-20
T100 S101 B100         18 19-JAN-20 15-FEB-20
T100 S101 B102         19 11-JAN-20 11-FEB-20           3
T101 S101 B102         19 13-JAN-20 12-FEB-20           2

8 rows selected.

SQL>
SQL> DELETE FROM Boat WHERE BID='B100';

1 row deleted.

SQL> DELETE FROM Sailor WHERE SID='S101';

1 row deleted.

SQL>
SQL> SELECT * FROM Reservations;

T_ID S_ID B_ID        NOP D_RES     D_SAIL    NO_CHILDREN
---- ---- ---- ---------- --------- --------- -----------
T101 S103 B102         17 14-JAN-20 15-FEB-20
T102 S105 B102         18 15-JAN-20 15-FEB-20
T101 S104 B101         18 17-JAN-20 15-FEB-20

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> spool off;
```

```
SQL> D:/01-College/databaselab/sql_files/ex2.sql
SP2-0734: unknown command beginning "D:/01-Coll..." - rest of line
ignored.
SQL> @D:/01-College/databaselab/sql_files/ex2.sql
SQL> REM Dropping table
SQL>
SQL> DROP TABLE PC;

Table dropped.

SQL>
SQL> REM Creating table PC
SQL>
SQL> CREATE TABLE PC(
  2        MODEL number(4),
  3        SPEED number(4),
  4        RAM number(3),
  5        HD number(2),
  6        RD varchar2(6),
  7        PRICE number(4)
  8  )
  9  /

Table created.

SQL>
SQL> REM Displaying table
SQL>
SQL> desc PC;
 Name
Null?    Type
 ----------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-------------------------------------------------------
 MODEL
NUMBER(4)
 SPEED
NUMBER(4)
 RAM
NUMBER(3)
 HD
NUMBER(2)
 RD
VARCHAR2(6)
 PRICE
NUMBER(4)

SQL>
SQL>
SQL> REM 1. Add the first row of data to the PC table from the above
sample data. Do not list the
SQL> REM columns in the INSERT clause.
SQL>
SQL> INSERT INTO PC VALUES(1001,1500,128,60,'12xDVD',2499);

1 row created.

SQL>
SQL> REM 2. Populate the table with the remaining sample data. This time,
list the columns
```

```
SQL> REM explicitly in the INSERT clause.
SQL>
SQL> INSERT INTO PC(MODEL,SPEED,RAM,HD,RD,PRICE)
VALUES(1002,866,128,20,'8xDVD',1999);

1 row created.

SQL> INSERT INTO PC(MODEL,SPEED,RAM,HD,RD,PRICE)
VALUES(1003,1000,128,40,'12xDVD',1499);

1 row created.

SQL> INSERT INTO PC(MODEL,SPEED,RAM,HD,RD,PRICE)
VALUES(1004,1200,128,80,'12xDVD',1699);

1 row created.

SQL> INSERT INTO PC(MODEL,SPEED,RAM,HD,RD,PRICE)
VALUES(1005,1300,256,80,'16xDVD',2299);

1 row created.

SQL>
SQL> REM Displaying table with values
SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        128         60 12xDVD       2499
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
      1004       1200        128         80 12xDVD       1699
      1005       1300        256         80 16xDVD       2299

SQL>
SQL>
SQL> REM 3. Change the RAM speed of 1001 model to 256
SQL>
SQL> UPDATE PC SET RAM=256 WHERE MODEL=1001;

1 row updated.

SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 12xDVD       2499
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
      1004       1200        128         80 12xDVD       1699
      1005       1300        256         80 16xDVD       2299

SQL>
SQL>
SQL> REM 4. Due to increase in the price of processor, hike the price of
PC with speed greater than
SQL> REM 1000 by 2%. Verify your changes to the table.
SQL>
```

```
SQL> REM BEFORE UPDATE
SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 12xDVD       2499
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
      1004       1200        128         80 12xDVD       1699
      1005       1300        256         80 16xDVD       2299

SQL>
SQL> UPDATE PC SET PRICE=PRICE*1.02 WHERE (SPEED>1000);

3 rows updated.

SQL>
SQL> REM AFTER UPDATE
SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 12xDVD       2549
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
      1004       1200        128         80 12xDVD       1733
      1005       1300        256         80 16xDVD       2345

SQL>
SQL>
SQL> REM 5.Mark an intermediate point in the processing of the
transaction.
SQL>
SQL> SAVEPOINT save_point_intermediate;

Savepoint created.

SQL>
SQL>
SQL> REM 6. Change the speed of RD to 16x for the PC model with speed
atleast 1000MHz and
SQL> REM having 128 RAM or atleast 40 GB HD.
SQL>
SQL> REM BEFORE UPDATE
SQL>
SQL> SELECT * FROM PC
  2
SQL> UPDATE PC SET RD='16x' WHERE (SPEED>=1000 AND (RAM=128 OR HD>=40));

4 rows updated.

SQL>
SQL> REM AFTER UPDATE
SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
```

```
      1001       1500        256         60 16x          2549
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 16x          1499
      1004       1200        128         80 16x          1733
      1005       1300        256         80 16x          2345

SQL>
SQL>
SQL> REM 7. Delete 1002 model from PC table.
SQL>
SQL> DELETE FROM PC WHERE MODEL=1002;

1 row deleted.

SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 16x          2549
      1003       1000        128         40 16x          1499
      1004       1200        128         80 16x          1733
      1005       1300        256         80 16x          2345

SQL>
SQL>
SQL> REM 8. Discard the changes performed in 6 and 7 without discarding
the earlier
SQL> REM operation(s).
SQL>
SQL> ROLLBACK TO save_point_intermediate;

Rollback complete.

SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 12xDVD       2549
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
      1004       1200        128         80 12xDVD       1733
      1005       1300        256         80 16xDVD       2345

SQL>
SQL> REM 9.Commit the changes.
SQL>
SQL> COMMIT;

Commit complete.

SQL>
SQL> SELECT * FROM PC;

     MODEL      SPEED        RAM         HD RD          PRICE
---------- ---------- ---------- ---------- ------ ----------
      1001       1500        256         60 12xDVD       2549
      1002        866        128         20 8xDVD        1999
      1003       1000        128         40 12xDVD       1499
```

```
      1004       1200        128         80 12xDVD       1733
      1005       1300        256         80 16xDVD       2345

SQL>
SQL>
SQL>
SQL> @D:/01-College/databaselab/sql_files/audio.sql
SQL> REM
**********************************************************************
SQL> REM Drop if any existing relation AUDIO
SQL> DROP TABLE AUDIO;

Table dropped.

SQL>
SQL> REM
**********************************************************************
SQL> REM Create the AUDIO relation to hold information about different
SQL> REM music albums
SQL>
SQL> CREATE TABLE AUDIO(
  2        audio_id number(3) constraint pk_aid primary key,title
varchar2(25),
  3        no_of_tracks number(1),
  4        release_date date,
  5        genre char(3) constraint ch_genre CHECK(genre
IN('CAR','DIV','MOV')),
  6        music varchar2(23),
  7        studio varchar2(10),
  8        price number(3)
  9  );

Table created.

SQL>
SQL>
SQL>
SQL> REM
**********************************************************************
SQL> REM Populate the AUDIO relation
SQL> REM
SQL> REM audio
(audio_id,title,no_of_tracks,release_date,genre,music,studio,price)
SQL>
SQL>
SQL> insert into audio values(100,'Nayagan',5,'21-OCT-
87','MOV','Ilaiyaraja','Pyramid',85);

1 row created.

SQL> insert into audio values(101,'Agni Nakshatram',6,'15-APR-
88','MOV','Ilaiyaraja','Pyramid',85);

1 row created.

SQL> insert into audio values(102,'Geethanjali',7,'19-MAY-
89','MOV','Ilaiyaraja','Lahari',90);

1 row created.
```

```
SQL> insert into audio values(103,'Anjali',7,'03-DEC-
90','MOV','Ilaiyaraja','Lahari',80);

1 row created.

SQL> insert into audio values(104,'Bombay',8,'11-MAR-95','MOV','AR
Rahman','MagnaSound',150);

1 row created.

SQL> insert into audio values(105,'Thalapathi',7,'05-NOV-
91','MOV','Ilaiyaraja','Lahari',100);

1 row created.

SQL> insert into audio values(106,'Roja',6,'11-MAY-92','MOV','AR
Rahman','MagnaSound',125);

1 row created.

SQL> insert into audio values(107,'Nadhamrutham',5,'01-NOV-
98','CAR',null,'SaReGaMa',150);

1 row created.

SQL> insert into audio values(108,'Gentleman',5,'30-JUL-93','MOV','AR
Rahman','Pyramid',100);

1 row created.

SQL> insert into audio values(109,'Thiruda Thiruda',6,'11-NOV-
93','MOV','AR Rahman','SaReGaMa',100);

1 row created.

SQL> insert into audio values(110,'Indian',6,'01-MAY-96','MOV','AR
Rahman','Pyramid',150);

1 row created.

SQL> insert into audio values(111,'Jeans',6,'01-APR-98','MOV','AR
Rahman','T-Series',150);

1 row created.

SQL> insert into audio values(112,'Rangeela',8,'08-SEP-95','MOV','AR
Rahman','T-Series',120);

1 row created.

SQL> insert into audio values(113,'Divine Collections',5,'21-OCT-
96','DIV','Kunnakudi Vaidhyanathan','SaReGaMa',175);

1 row created.

SQL> insert into audio values(114,'Krishnarpanam',6,'25-AUG-
97','DIV',null,'SaReGaMa',175);

1 row created.
```

```
SQL>
SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 1. Select the audio id, title and release date that was released
after the year 1997.
SQL>
SQL> SELECT audio_id,title,release_date
  2  FROM audio
  3  WHERE EXTRACT(YEAR FROM release_date)>1997;

  AUDIO_ID TITLE                     RELEASE_D
---------- ------------------------- ---------
       107 Nadhamrutham              01-NOV-98
       111 Jeans                     01-APR-98

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 2. Display the title, release date, hike the price by 10% for
the audio released after 1995
SQL> REM and label the column as Audio Name and New Price respectively.
SQL>
SQL> SELECT title as "Audio Name",release_date,price*1.10 as "New Price"
  2  FROM audio
  3  WHERE EXTRACT(YEAR FROM release_date)>1995;

Audio Name                RELEASE_D  New Price
------------------------- --------- ----------
Nadhamrutham              01-NOV-98        165
Indian                    01-MAY-96        165
Jeans                     01-APR-98        165
Divine Collections        21-OCT-96      192.5
Krishnarpanam             25-AUG-97      192.5

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 3. Display the unique studio from AUDIO relation.
SQL>
SQL> SELECT DISTINCT(studio) FROM audio;

STUDIO
----------
T-Series
Pyramid
Lahari
SaReGaMa
MagnaSound

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 4. Show the title, release date, genre, studio and price of MOV
or DIV type audio, but
SQL> REM not released by Pyramid studio.
```

```
SQL>
SQL> SELECT title,release_date,genre,studio,price
  2  FROM audio
  3  WHERE genre IN('MOV','DIV') AND NOT studio='Pyramid';

TITLE                     RELEASE_D GEN STUDIO          PRICE
------------------------- --------- --- ---------- ----------
Geethanjali               19-MAY-89 MOV Lahari             90
Anjali                    03-DEC-90 MOV Lahari             80
Bombay                    11-MAR-95 MOV MagnaSound        150
Thalapathi                05-NOV-91 MOV Lahari            100
Roja                      11-MAY-92 MOV MagnaSound        125
Thiruda Thiruda           11-NOV-93 MOV SaReGaMa          100
Jeans                     01-APR-98 MOV T-Series          150
Rangeela                  08-SEP-95 MOV T-Series          120
Divine Collections        21-OCT-96 DIV SaReGaMa          175
Krishnarpanam             25-AUG-97 DIV SaReGaMa          175

10 rows selected.

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 5. Display the audio id, title, music and price of audios either
in the range of Rs.100 to
SQL> REM Rs.150 or music by AR Rahman.
SQL>
SQL> SELECT audio_id,title,music,price
  2  FROM audio
  3  WHERE (price BETWEEN 100 AND 150) OR music='AR Rahman';

  AUDIO_ID TITLE                     MUSIC                       PRICE
---------- ------------------------- ----------------------- ----------
       104 Bombay                    AR Rahman                      150
       105 Thalapathi                Ilaiyaraja                     100
       106 Roja                      AR Rahman                      125
       107 Nadhamrutham                                             150
       108 Gentleman                 AR Rahman                      100
       109 Thiruda Thiruda           AR Rahman                      100
       110 Indian                    AR Rahman                      150
       111 Jeans                     AR Rahman                      150
       112 Rangeela                  AR Rahman                      120

9 rows selected.

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 6. Display the audio title, number of tracks, release date,
musician and studio that was
SQL> REM released during 1995.
SQL>
SQL> SELECT title,no_of_tracks,release_date,music,studio
  2  FROM audio
  3  WHERE EXTRACT(YEAR FROM  release_date)=1995;

TITLE                     NO_OF_TRACKS RELEASE_D MUSIC
STUDIO
```

```
------------------------- ------------ --------- -----------------------
----------
Bombay                                8 11-MAR-95 AR Rahman
MagnaSound
Rangeela                              8 08-SEP-95 AR Rahman
T-Series

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 7. Display the title, genre, musician, studio and price of an
audio for which the music
SQL> REM was scored and the title has either a as second letter or starts
with N.
SQL>
SQL> SELECT title,genre,music,studio,price
  2  FROM audio
  3  WHERE (title LIKE '_a%' OR title LIKE 'N%') AND music IS NOT NULL;

TITLE                     GEN MUSIC                   STUDIO
PRICE
------------------------- --- ----------------------- ---------- --------
--
Nayagan                   MOV Ilaiyaraja              Pyramid
85
Rangeela                  MOV AR Rahman               T-Series
120

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 8. Display the audio details like title, number of tracks,
genre, music and studio that
SQL> REM has atmost 6 tracks by SaReGaMa Studio. If the music is not
scored by any musician,
SQL> REM display as No Music.
SQL>
SQL> SELECT title,no_of_tracks,genre,NVL(music,'No Music') AS
MUSIC,studio
  2  FROM audio
  3  WHERE no_of_tracks<=6 AND studio='SaReGaMa';

TITLE                     NO_OF_TRACKS GEN MUSIC                   STUDIO
------------------------- ------------ --- ----------------------- ------
----
Nadhamrutham                         5 CAR No Music
SaReGaMa
Thiruda Thiruda                      6 MOV AR Rahman
SaReGaMa
Divine Collections                   5 DIV Kunnakudi Vaidhyanathan
SaReGaMa
Krishnarpanam                        6 DIV No Music
SaReGaMa

SQL>
SQL>
SQL> REM
**********************************************************************
```

```
SQL>
SQL> REM 9. Display the title and genre of the album that starts with B,
N, K or R, but not a CAR
SQL> REM genre.
SQL>
SQL> SELECT title,genre
  2  FROM audio
  3  WHERE (title LIKE 'B%' OR title LIKE 'N%' OR title LIKE 'K%' OR
title LIKE 'R%') AND NOT genre='CAR';

TITLE                     GEN
------------------------- ---
Nayagan                   MOV
Bombay                    MOV
Roja                      MOV
Rangeela                  MOV
Krishnarpanam             DIV

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 10. List the album title, Date of Release, number of tracks,
studio and musician that was
SQL> REM released before 30 years and 3 months.
SQL>
SQL> SELECT title,release_date,no_of_tracks,studio,music
  2  FROM audio
  3  WHERE (EXTRACT(YEAR FROM sysdate)-EXTRACT(YEAR FROM
release_date))>30 AND release_date<=ADD_MONTHS(TRUNC(sysdate),-3);

TITLE                     RELEASE_D NO_OF_TRACKS STUDIO     MUSIC
------------------------- --------- ------------ ---------- -------------
----------
Nayagan                   21-OCT-87            5 Pyramid    Ilaiyaraja
Agni Nakshatram           15-APR-88            6 Pyramid    Ilaiyaraja
Geethanjali               19-MAY-89            7 Lahari     Ilaiyaraja
Anjali                    03-DEC-90            7 Lahari     Ilaiyaraja

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 11. List the title, number of tracks, release date, genre,
music, studio and price of audio
SQL> REM for which AR Rahman had scored the music from 1995 to 1998 in
alphabetical order
SQL> REM by title.
SQL>
SQL> SELECT title,no_of_tracks,release_date,genre,music,studio,price
  2  FROM audio
  3  WHERE EXTRACT(YEAR FROM release_date) BETWEEN 1995 AND 1998 AND
music='AR Rahman'
  4  ORDER BY title asc;

TITLE                     NO_OF_TRACKS RELEASE_D GEN MUSIC
STUDIO          PRICE
------------------------- ------------ --------- --- --------------------
--- ---------- ----------
```

```
Bombay                                  8 11-MAR-95 MOV AR Rahman
MagnaSound        150
Indian                                  6 01-MAY-96 MOV AR Rahman
Pyramid           150
Jeans                                   6 01-APR-98 MOV AR Rahman
T-Series          150
Rangeela                                8 08-SEP-95 MOV AR Rahman
T-Series          120

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 12. How many movie audio(s) have a name that ends with letter a.
SQL>
SQL> SELECT COUNT(title) AS AUDIO_COUNT
  2  FROM audio
  3  WHERE title like '%a' AND genre='MOV';

AUDIO_COUNT
-----------
          3

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 13. Display the maximum, minimum and average price of movie
audio.
SQL>
SQL> SELECT MAX(price) AS MAX_PRICE,MIN(price) AS MIN_PRICE,AVG(price) AS
AVG_PRICE
  2  FROM audio
  3  WHERE genre='MOV';

 MAX_PRICE  MIN_PRICE  AVG_PRICE
---------- ---------- ----------
       150         80     111.25

SQL>
SQL> REM
**********************************************************************
SQL>
SQL> REM 14. Show the cheapest, costlier, average and total price of
audio for studiowise.
SQL> REM Label the columns Cheaper, Costly, Average, and Total
respectively. Round your results to
SQL> REM the nearest whole number. Sort your result by alphabetical order
of studio.
SQL>
SQL> SELECT studio,MIN(price) AS CHEAPER,MAX(price) AS
COSTLY,ROUND(AVG(price)) AS AVERAGE,SUM(price) AS TOTAL
  2  FROM audio
  3  GROUP BY studio
  4  ORDER BY studio asc;

STUDIO        CHEAPER     COSTLY    AVERAGE      TOTAL
---------- ---------- ---------- ---------- ----------
Lahari             80        100         90        270
MagnaSound        125        150        138        275
```

```
Pyramid            85        150        105        420
SaReGaMa          100        175        150        600
T-Series          120        150        135        270

SQL>
SQL> REM
**************************************************************************
SQL>
SQL> REM 15. List the studio which sells the audio for a minimum price of
atleast Rs.100.
SQL>
SQL> SELECT studio,MIN(price)
  2  FROM audio
  3  HAVING MIN(price)>=100
  4  GROUP BY studio;

STUDIO     MIN(PRICE)
---------- ----------
T-Series          120
SaReGaMa          100
MagnaSound        125

SQL>
SQL> REM
**************************************************************************
SQL>
SQL> REM 16. For each studio, show the musician name and number of times
a musician scored
SQL> REM the music. Include only the audio that was released with in 30
years. Exclude any
SQL> REM groups if a musician scored music at most once. Sort the output
by studio name in
SQL> REM descending order.
SQL>
SQL> SELECT music,COUNT(music) AS no_times,studio
  2  FROM audio
  3  WHERE EXTRACT(YEAR FROM sysdate)-EXTRACT(YEAR FROM release_date)<=30
  4  GROUP BY studio,music
  5  HAVING COUNT(MUSIC)>1
  6  ORDER BY studio desc;

MUSIC                    NO_TIMES STUDIO
----------------------- ---------- ----------
AR Rahman                        2 T-Series
AR Rahman                        2 Pyramid
AR Rahman                        2 MagnaSound

SQL>
SQL> REM
**************************************************************************
SQL>
SQL>
SQL>
SQL> spool off;
```

```
SQL> @Z:\1106\carqfin.sql
SQL> REM 1. Display the models that was not manufactured by any of the
car makers
SQL>
SQL> SELECT model,fullname
  2  FROM MODEL_DETAILS,CAR_MAKERS
  3  WHERE MODEL_DETAILS.maker = CAR_MAKERS.Id
  4  AND MODEL_DETAILS.model NOT IN(SELECT model FROM CAR_NAMES);

MODEL                                     FULLNAME
----------------------------------------- ------------------------------
kia                                       Kia Motors
hyundai                                   Hyundai
jeep                                      Chrysler
scion                                     Toyota

SQL>
SQL> REM
**************************************************************************
**************************
SQL>
SQL> REM 2. For all the continents list the number of car makers if there
were a car manufacturing company
SQL>
SQL> SELECT c1.Continent,count(c3.Id) AS No_of_makers
  2  FROM CONTINENTS c1 LEFT OUTER JOIN COUNTRIES c2
  3  ON c1.ContId = c2.Continent
  4  LEFT OUTER JOIN  CAR_MAKERS c3
  5  ON C2.CountryId=c3.Country
  6  GROUP BY c1.Continent;

CONTINENT  NO_OF_MAKERS
---------- ------------
europe               11
africa                0
america               4
asia                  7
australia             0

SQL>
SQL> REM
**************************************************************************
**************************
SQL>
SQL> REM 3. Display the pair of cars (ID) that has same mileage,
horsepower and acceleration. The pairs should not be repeated in the
result.
SQL>
SQL> REM USING SUBQUERY
SQL>
SQL> SELECT C1.Id,C2.Id
  2  FROM CAR_DETAILS C1,CAR_DETAILS C2
  3  WHERE C1.mpg=C2.mpg AND C1.horsepower=C2.horsepower AND
C1.accel=C2.accel
  4  AND C1.Id<C2.Id;

        ID         ID
---------- ----------
         9         20
        25         36
```

```
        80         97
       105        143
       126        155

SQL>
SQL> REM USING JOIN
SQL>
SQL> SELECT C1.Id,C2.Id
  2  FROM CAR_DETAILS C1 JOIN  CAR_DETAILS C2
  3  ON(C1.mpg=C2.mpg AND C1.horsepower=C2.horsepower AND
C1.accel=C2.accel)
  4  AND C1.Id < C2.Id;

        ID         ID
---------- ----------
         9         20
        25         36
        80         97
       105        143
       126        155

SQL>
SQL> REM
********************************************************************************
*************************
SQL>
SQL> REM 4. Display the number of cars produced by each car manufacturing
company with in each model. Sort the result by the company name.
SQL>
SQL> SELECT cm.fullname,md.Model,count(cn.Id)
  2  FROM CAR_MAKERS cm,MODEL_DETAILS md,CAR_NAMES cn
  3  WHERE cm.Id=md.maker AND md.Model=cn.model
  4  GROUP BY cm.fullname,md.model
  5  ORDER BY cm.fullname;

FULLNAME                       MODEL
COUNT(CN.ID)
------------------------------ ---------------------------------------- -
-----------
American Motor Company         amc
29
BMW                            bmw
2
Chrysler                       chrysler
6
Chrysler                       dodge
28
Chrysler                       plymouth
32
Citroen                        citroen
1
Daimler Benz                   mercedes
1
Daimler Benz                   mercedes-benz
2
Fiat                           fiat
8
Ford Motor Company             capri
1
```

```
Ford Motor Company              ford
53
Ford Motor Company              mercury
11
General Motors                  buick
17
General Motors                  cadillac
2
General Motors                  chevrolet
48
General Motors                  oldsmobile
10
General Motors                  pontiac
16
Honda                           honda
13
Mazda                           mazda
12
Nissan Motors                   datsun
23
Nissan Motors                   nissan
1
Opel                            opel
4
Peugeaut                        peugeot
8
Renault                         renault
5
Saab                            saab
5
Subaru                          subaru
4
Toyota                          toyota
26
Triumph                         triumph
1
Volkswagen                      audi
7
Volkswagen                      volkswagen
23
Volvo                           volvo
6
hi                              hi
1

32 rows selected.

SQL>
SQL> REM
********************************************************************************
**************************
SQL>
SQL> REM 5. Display the model, name of car, mpg and weight of car(s) with
maximum mileage among the heavy weight (bulky) cars. The car with weight
more than the REM average weight of all cars are known as heavy weight
(bulky) cars.
SQL>
SQL> REM USING JOIN
SQL>
SQL> SELECT Id,model,descr,mpg,weight
```

```
  2  FROM CAR_NAMES JOIN CAR_DETAILS USING(Id)
  3  WHERE mpg=(SELECT MAX(mpg) FROM CAR_DETAILS WHERE weight>(SELECT
AVG(weight) FROM CAR_DETAILS)) AND weight>(SELECT AVG(weight) FROM
CAR_DETAILS);

        ID MODEL                                          DESCR
MPG     WEIGHT
---------- ---------------------------------------- --------------------
---------------------------- ---------- ----------
       396 oldsmobile                                    oldsmobile cutlass
ciera (diesel)                           38       3015

SQL>
SQL>
SQL> REM
********************************************************************************
*************************
SQL>
SQL> REM 6. Display the details
(model,car_name,mileage,horsepower,acceleration,weight) of car(s) having
mileage, horsepower, acceleration more than the average REM of mpg,
horsepower, accel of all cars and its weight should be lesser than the
average weight of all cars.
SQL>
SQL> SELECT model,descr,mpg,horsepower,accel,weight
  2  FROM CAR_NAMES JOIN CAR_DETAILS
  3  USING(Id)
  4  WHERE mpg>(SELECT AVG(mpg) FROM CAR_DETAILS) AND horsepower>(SELECT
AVG(horsepower) FROM CAR_DETAILS) AND
  5  accel>(SELECT AVG(accel) FROM CAR_DETAILS) AND weight<(SELECT
AVG(weight) FROM CAR_DETAILS);

MODEL                                     DESCR
MPG HORSEPOWER      ACCEL     WEIGHT
---------------------------------------- --------------------------------
----------------- ---------- ---------- ---------- ----------
buick                                     buick century limited
25        110       16.4       2945

SQL>
SQL> REM
********************************************************************************
*************************
SQL>
SQL> REM 7. List the year, car maker that manufactured maximum number of
cars.
SQL>
SQL> SELECT C3.year, C1.maker, COUNT(*) AS CARS
  2  FROM CAR_MAKERS C1, MODEL_DETAILS M, CAR_NAMES C2, CAR_DETAILS C3
  3  WHERE C1.Id = M.maker
  4  AND M.model = C2.model
  5  AND C2.Id = C3.Id
  6  GROUP BY C3.year, C1.maker
  7  HAVING COUNT(*) = (
  8  SELECT MAX(COUNT(*))
  9  FROM CAR_MAKERS C1, MODEL_DETAILS M, CAR_NAMES C2, CAR_DETAILS C3
 10  WHERE C1.Id = M.maker
 11  AND M.model = C2.model
 12  AND C2.Id = C3.Id
 13  GROUP BY C3.year, C1.maker
```

```
 14  )
 15  ORDER BY C3.year;

      YEAR MAKER                CARS
---------- --------------- ----------
      1973 gm                      11

SQL>
SQL> REM
********************************************************************************
*************************
SQL>
SQL> REM 8. Display the maker name, model name, car name, mileage and
year of the car with the maximum mileage for each model having more than
one car.
SQL> REM Sort the result by the car maker.
SQL>
SQL> SELECT cm.fullName,mdo.model,cn.descr "CAR NAME" ,cd.mpg,cd.year
  2  FROM CAR_MAKERS cm , MODEL_DETAILS mdo, CAR_NAMES cn,CAR_DETAILS cd
  3  WHERE cm.Id=mdo.maker AND mdo.model=cn.model AND cd.Id=cn.Id
  4  AND cd.mpg=(SELECT MAX(cd.mpg)
  5  FROM MODEL_DETAILS md, CAR_NAMES cn,CAR_DETAILS cd
  6  WHERE md.model=cn.model AND cd.Id=cn.Id AND md.model=mdo.model
  7  GROUP BY md.model
  8  HAVING COUNT(*)>1)
  9  ORDER BY cm.fullName;

FULLNAME                       MODEL
CAR NAME                                                MPG       YEAR
------------------------------ ------------------------------------------ -
------------------------------------------------- ---------- ----------
American Motor Company         amc
amc spirit dl                                          27.4       1979
BMW                            bmw
bmw 2002                                                 26       1970
Chrysler                       chrysler
chrysler lebaron medallion                               26       1982
Chrysler                       dodge
dodge charger 2.2                                        36       1982
Chrysler                       plymouth
plymouth champ                                           39       1981
Daimler Benz                   mercedes-benz
mercedes-benz 240d                                       30       1980
Fiat                           fiat
fiat strada custom                                     37.3       1979
Ford Motor Company             ford
ford fiesta                                            36.1       1978
Ford Motor Company             mercury
mercury lynx l                                           36       1982
General Motors                 pontiac
pontiac phoenix                                        33.5       1979
General Motors                 cadillac
cadillac eldorado                                        23       1979
General Motors                 chevrolet
chevrolet cavalier 2-door                                34       1982
General Motors                 oldsmobile
oldsmobile cutlass ciera (diesel)                        38       1982
General Motors                 buick
buick opel isuzu deluxe                                  30       1977
```

```
Honda                          honda
honda civic 1500 gl                                    44.6       1980
Mazda                          mazda
mazda glc                                              46.6       1980
Nissan Motors                  datsun
datsun 210                                             40.8       1980
Opel                           opel
opel 1900                                                28       1971
Peugeaut                       peugeot
peugeot 304                                              30       1971
Renault                        renault
renault lecar deluxe                                   40.9       1980
Saab                           saab
saab 99le                                                25       1975
Saab                           saab
saab 99e                                                 25       1970
Subaru                         subaru
subaru dl                                              33.8       1980
Toyota                         toyota
toyota starlet                                         39.1       1981
Volkswagen                     volkswagen
vw rabbit c (diesel)                                   44.3       1980
Volkswagen                     audi
audi 5000s (diesel)                                    36.4       1980
Volvo                          volvo
volvo diesel                                           30.7       1981

27 rows selected.

SQL>
SQL> REM
********************************************************************************
**************************
SQL>
SQL> REM 9. Rewrite the query 1.
SQL>
SQL> REM USING SET OPERATIONS
SQL>
SQL> SELECT MODEL FROM MODEL_DETAILS WHERE MAKER IN (SELECT maker
  2  FROM MODEL_DETAILS
  3  INTERSECT
  4  SELECT id
  5  FROM CAR_MAKERS) AND model NOT IN(Select Model FROM CAR_NAMES);

MODEL
----------------------------------------
kia
hyundai
jeep
scion

SQL>
SQL>
SQL> SELECT MODEL
  2  FROM MODEL_DETAILS md , CAR_MAKERS cm
  3  WHERE md.maker = cm.Id
  4  AND (md.model NOT IN(SELECT model FROM CAR_NAMES));

MODEL
----------------------------------------
```

```
kia
hyundai
jeep
scion

SQL>
SQL> REM
**************************************************************************
*************************
SQL>
SQL> REM 10. List the car names (description) and its details that was
manufactured on 1976 and 1982.
SQL>
SQL> SELECT
md.model,descr,cd.Id,mpg,cylinders,edispl,horsepower,weight,accel,year
  2  FROM CAR_NAMES cn, CAR_DETAILS cd, MODEL_DETAILS md
  3  WHERE cn.model=md.model and cd.id=cn.id and descr in((
  4  SELECT cn.descr
  5  FROM CAR_NAMES cn, CAR_DETAILS cd, MODEL_DETAILS md
  6  WHERE cn.model=md.model and cd.Id=cn.Id
  7  GROUP BY cn.descr,year
  8  HAVING YEAR=1976)
  9  INTERSECT
 10  (SELECT cn.descr
 11  FROM CAR_NAMES cn, CAR_DETAILS cd, MODEL_DETAILS md
 12  WHERE cn.model=md.model and cd.Id=cn.Id
 13  GROUP BY cn.descr,year
 14  HAVING YEAR=1982))
 15  AND year in(1976,1982);

MODEL                                          DESCR
ID         MPG  CYLINDERS     EDISPL HORSEPOWER     WEIGHT      ACCEL
YEAR
---------------------------------------- ---------------------------------
---------------- ---------- ---------- ---------- ---------- ----------
---------- ---------- ----------
honda                                          honda civic
392         38          4         91         67       1965         15
1982
honda                                          honda civic
206         33          4         91         53       1795       17.4
1976
toyota                                         toyota corolla
391         34          4        108         70       2245       16.9
1982
toyota                                         toyota corolla
213         28          4         97         75       2155       16.4
1976

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
```

```
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> spool off;
```

```
SQL> @D:\01-College\databaselab\sql_files\ex4.sql
SQL> set echo on
SQL> set linesize 300
SQL> set pagesize 100
SQL>
SQL> REM
**************************************************************
SQL>
SQL> REM 1. Create a view named Datsun_Cars, which display the car id,
model and
SQL> REM descriptions of Datsun model.
SQL>
SQL>
SQL> REM Creating View and displaying
SQL>
SQL> CREATE OR REPLACE VIEW Datsun_Cars(carid,model,description)
  2  AS SELECT Id,model,descr
  3  FROM CAR_NAMES
  4  WHERE model='datsun';

View created.

SQL>
SQL> DESC Datsun_Cars;
 Name
Null?    Type
 ----------------------------------------------------------------------------
----------------------------------------------------------------------------
---------------------------- -------- -------------------------------
----------------------------------------------------------------------------
--------
 CARID
NOT NULL NUMBER
 MODEL
VARCHAR2(40)
 DESCRIPTION
VARCHAR2(50)

SQL>
SQL> REM ------------------------------------------------------------------
-
>
SQL> REM Displaying the rows in the view
SQL>
SQL> SELECT * FROM Datsun_Cars;

     CARID MODEL                                     DESCRIPTION
---------- ---------------------------------------- --------------------
------------------------------
        25 datsun                                   datsun pl510
        36 datsun                                   datsun pl510
        62 datsun                                   datsun 1200
        89 datsun                                   datsun 510 (sw)
       118 datsun                                   datsun 610
       137 datsun                                   datsun b210
       153 datsun                                   datsun 710
       181 datsun                                   datsun 710
       212 datsun                                   datsun b-210
       228 datsun                                   datsun f-10 hatchback
       249 datsun                                   datsun 810
```

```
       255 datsun                                          datsun b210 gx
       276 datsun                                          datsun 510
       281 datsun                                          datsun 200-sx
       311 datsun                                          datsun 210
       320 datsun                                          datsun 310
       328 datsun                                          datsun 510 hatchback
       332 datsun                                          datsun 210
       341 datsun                                          datsun 280-zx
       355 datsun                                          datsun 210 mpg
       365 datsun                                          datsun 200sx
       371 datsun                                          datsun 810 maxima
       394 datsun                                          datsun 310 gx

23 rows selected.

SQL>
SQL> SELECT * FROM USER_UPDATABLE_COLUMNS WHERE TABLE_NAME='DATSUN_CARS';

OWNER
TABLE_NAME
------------------------------------------------------------------------------
------------------------------------------------------------ ----------------
------------------------------------------------------------------------------
----------------------------------------
COLUMN_NAME
UPD INS DEL
------------------------------------------------------------------------------
------------------------------------------------------------ --- --- ---
SYSTEM
DATSUN_CARS
CARID
YES YES YES

SYSTEM
DATSUN_CARS
MODEL
YES YES YES

SYSTEM
DATSUN_CARS
DESCRIPTION
YES YES YES


SQL>
SQL> REM ------------------------------------------------------------------
-
>
SQL> SAVEPOINT S1;

Savepoint created.

SQL>
SQL> REM Inserting into view
SQL>
SQL> INSERT INTO Datsun_Cars VALUES (500,'datsun','Wxyz');

1 row created.

SQL>
```

```
SQL> SELECT * FROM CAR_NAMES WHERE Id=500 AND model='datsun' AND
descr='Wxyz';

        ID MODEL                                    DESCR
---------- ---------------------------------------- --------------------
------------------------------
       500 datsun                                   Wxyz

SQL>
SQL> REM INSERTION IS POSSIBLE
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Updating values in the table
SQL>
SQL> UPDATE Datsun_Cars SET description='datsun xyz' WHERE carid=25;

1 row updated.

SQL>
SQL> SELECT * FROM CAR_NAMES WHERE Id=25;

        ID MODEL                                    DESCR
---------- ---------------------------------------- --------------------
------------------------------
        25 datsun                                   datsun xyz

SQL>
SQL> REM UPDATION IS POSSIBLE
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Deleting value from table
SQL>
SQL> DELETE FROM Datsun_Cars WHERE carid=500;

1 row deleted.

SQL>
SQL> SELECT * FROM CAR_NAMES WHERE Id=500;

no rows selected

SQL>
SQL> REM DELETION IS POSSIBLE
SQL>
SQL> REM CONCLUSION: INSERTION,UPDATION,DELETION IS POSSIBLE
SQL>
SQL> ROLLBACK TO SAVEPOINT S1;

Rollback complete.

SQL>
SQL> REM
**************************************************************
SQL>
SQL> REM 2. Create a view called Car_List that shows the car id, model,
description
```

```
SQL> REM and operational parameters of all cars produced during 1974.
Make sure
SQL> REM that, the year should not be reassigned to any other value
through view.
SQL>
SQL> REM Creating View and displaying
SQL>
SQL> CREATE OR REPLACE VIEW
Car_List(carid,model,description,mpg,cylinders,edispl,horsepower,weight,a
ccel,year)
  2  AS SELECT * FROM CAR_NAMES cn JOIN CAR_DETAILS cd
  3  USING(Id)
  4  WHERE cd.year=1974
  5  WITH CHECK OPTION;

View created.

SQL>
SQL> DESC Car_List;
 Name
Null?    Type
 ----------------------------------------------------------------------------
----------------------------------------------------------------------------
----------------------------- -------- -------------------------------------
----------------------------------------------------------------------------
--------
 CARID
NOT NULL NUMBER
 MODEL
VARCHAR2(40)
 DESCRIPTION
VARCHAR2(50)
 MPG
NUMBER
 CYLINDERS
NUMBER
 EDISPL
NUMBER
 HORSEPOWER
NUMBER
 WEIGHT
NUMBER
 ACCEL
NUMBER(3,1)
 YEAR
NUMBER

SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Displaying the rows in the view
SQL>
SQL> SELECT * FROM Car_List;

     CARID MODEL                                     DESCRIPTION
MPG  CYLINDERS     EDISPL HORSEPOWER     WEIGHT      ACCEL       YEAR
---------- ---------------------------------------- --------------------
----------------------------- ---------- ---------- ---------- ----------
---------- ---------- ----------
```

```
       133 plymouth                                                plymouth duster
20          6        198         95       3102        16.5       1974
       134 ford                                                    ford maverick
21          6        200                  2875           17       1974
       135 amc                                                     amc hornet
19          6        232        100       2901           16       1974
       136 chevrolet                                               chevrolet nova
15          6        250        100       3336           17       1974
       137 datsun                                                  datsun b210
31          4         79         67       1950           19       1974
       138 ford                                                    ford pinto
26          4        122         80       2451        16.5        1974
       139 toyota                                                  toyota corolla 1200
32          4         71         65       1836           21       1974
       140 chevrolet                                               chevrolet vega
25          4        140         75       2542           17       1974
       141 chevrolet                                               chevrolet chevelle
malibu classic                                  16          6        250
100       3781         17       1974
       142 amc                                                     amc matador
16          6        258        110       3632           18       1974
       143 plymouth                                                plymouth satellite
sebring                                         18          6        225
105       3613       16.5       1974
       144 ford                                                    ford gran torino
16          8        302        140       4141           14       1974
       145 buick                                                   buick century luxus
(sw)                                            13          8        350
150       4699       14.5       1974
       146 dodge                                                   dodge coronet custom
(sw)                                       14          8        318        150
4457       13.5       1974
       147 ford                                                    ford gran torino (sw)
14          8        302        140       4638           16       1974
       148 amc                                                     amc matador (sw)
14          8        304        150       4257         15.5       1974
       149 audi                                                    audi fox
29          4         98         83       2219         16.5       1974
       150 volkswagen                                              volkswagen dasher
26          4         79         67       1963         15.5       1974
       151 opel                                                    opel manta
26          4         97         78       2300         14.5       1974
       152 toyota                                                  toyota corona
31          4         76         52       1649         16.5       1974
       153 datsun                                                  datsun 710
32          4         83         61       2003           19       1974
       154 dodge                                                   dodge colt
28          4         90         75       2125         14.5       1974
       155 fiat                                                    fiat 128
24          4         90         75       2108         15.5       1974
       156 fiat                                                    fiat 124 tc
26          4        116         75       2246           14       1974
       157 honda                                                   honda civic
24          4        120         97       2489           15       1974
       158 subaru                                                  subaru
26          4        108         93       2391         15.5       1974
       159 fiat                                                    fiat x1.9
31          4         79         67       2000           16       1974

27 rows selected.
```

```
SQL>
SQL> SELECT * FROM USER_UPDATABLE_COLUMNS WHERE TABLE_NAME='CAR_LIST';

OWNER
TABLE_NAME
------------------------------------------------------------------------------
------------------------------------------------------------ ------------------
------------------------------------------------------------------------------
--------------------------------------
COLUMN_NAME
UPD INS DEL
------------------------------------------------------------------------------
------------------------------------------------------ --- --- ---
SYSTEM
CAR_LIST
CARID
NO  NO  NO

SYSTEM
CAR_LIST
MODEL
YES YES YES

SYSTEM
CAR_LIST
DESCRIPTION
YES YES YES

SYSTEM
CAR_LIST
MPG
YES YES YES

SYSTEM
CAR_LIST
CYLINDERS
YES YES YES

SYSTEM
CAR_LIST
EDISPL
YES YES YES

SYSTEM
CAR_LIST
HORSEPOWER
YES YES YES

SYSTEM
CAR_LIST
WEIGHT
YES YES YES

SYSTEM
CAR_LIST
ACCEL
YES YES YES
```

```
SYSTEM
CAR_LIST
YEAR
YES YES YES


10 rows selected.

SQL>
SQL> SAVEPOINT S2;

Savepoint created.

SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Inserting into view
SQL>
SQL> INSERT INTO Car_List
VALUES(500,'datsun','p1510',25,3,97,65,2100,1974,14);
INSERT INTO Car_List VALUES(500,'datsun','p1510',25,3,97,65,2100,1974,14)
*
ERROR at line 1:
ORA-01733: virtual column not allowed here


SQL>
SQL> REM INSERTION IS NOT POSSIBLE
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Updating values in the table
SQL>
SQL> UPDATE Car_List SET description='xyzabc' WHERE carid=133;

1 row updated.

SQL>
SQL> SELECT * FROM Car_List WHERE carid=133;

     CARID MODEL                                          DESCRIPTION
MPG  CYLINDERS     EDISPL HORSEPOWER     WEIGHT      ACCEL       YEAR
---------- ---------- ---------- ---------- ---------- ----------
---------- ---------- ---------- ---------- ----------
       133 plymouth                                     xyzabc
20          6        198         95       3102       16.5       1974

SQL>
SQL> REM UPDATION IS POSSIBLE
SQL>
SQL> REM UPDATING YEAR WITH CHECK OPTION
SQL>
SQL> UPDATE Car_List SET YEAR=2001 WHERE carid=140;
UPDATE Car_List SET YEAR=2001 WHERE carid=140
       *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

```
SQL>
SQL> REM UPDATION ON YEAR IS NOT POSSIBLE DUE TO CHECK OPTION VIOLATION
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Deleting value from table
SQL>
SQL> DELETE FROM Car_List WHERE carid=159;
DELETE FROM Car_List WHERE carid=159
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.CD_FK) violated - child record
found


SQL>
SQL> REM DELETION IS NOT POSSIBLE
SQL>
SQL> ROLLBACK TO SAVEPOINT S2;

Rollback complete.

SQL>
SQL> REM CONCLUSION: INSERTION AND DELETION IS NOT POSSIBLE BUT UPDATION
IS POSSIBLE
SQL>
SQL> REM
****************************************************************
SQL>
SQL> REM 3. Create a view named European_Makers that will display the
maker
SQL> REM name, full name and the country name of car makers from Europe.
SQL>
SQL>
SQL> REM Creating View and displaying
SQL>
SQL> CREATE OR REPLACE VIEW European_Makers(maker,fullname,country) AS
  2  SELECT cm.maker,cm.fullname,co.countryname
  3  FROM CONTINENTS cn,COUNTRIES co,CAR_MAKERS cm
  4  WHERE cn.Contid=co.continent AND co.Countryid=cm.country
  5  AND cn.continent='europe';

View created.

SQL>
SQL> DESC European_Makers;
 Name
Null?    Type
 ----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------- -------- ------------------------------
----------------------------------------------------------------------
--------
 MAKER
VARCHAR2(15)
 FULLNAME
VARCHAR2(30)
```

```
 COUNTRY
VARCHAR2(20)

SQL>
SQL> REM ------------------------------------------------------------------
-
>
SQL> REM Displaying the rows in the view
SQL>
SQL> SELECT * FROM European_Makers;

MAKER           FULLNAME                       COUNTRY
--------------- ------------------------------ --------------------
volkswagen      Volkswagen                     germany
bmw             BMW                            germany
daimler benz    Daimler Benz                   germany
opel            Opel                           germany
citroen         Citroen                        france
peugeaut        Peugeaut                       france
renault         Renault                        france
fiat            Fiat                           italy
saab            Saab                           sweden
volvo           Volvo                          sweden
triumph         Triumph                        uk

11 rows selected.

SQL>
SQL> SELECT * FROM USER_UPDATABLE_COLUMNS WHERE
TABLE_NAME='EUROPEAN_MAKERS';

OWNER
TABLE_NAME
------------------------------------------------------------------------------
------------------------------------------------------------ ----------------
------------------------------------------------------------------------------
--------------------------------------
COLUMN_NAME
UPD INS DEL
------------------------------------------------------------------------------
------------------------------------------------------------ --- --- ---
SYSTEM
EUROPEAN_MAKERS
MAKER
YES YES YES

SYSTEM
EUROPEAN_MAKERS
FULLNAME
YES YES YES

SYSTEM
EUROPEAN_MAKERS
COUNTRY
NO  NO  NO


SQL>
SQL> SAVEPOINT S3;
```

```
Savepoint created.

SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Inserting into view
SQL>
SQL> INSERT INTO European_Makers VALUES ('kia','kia motors',10);
INSERT INTO European_Makers VALUES ('kia','kia motors',10)
*
ERROR at line 1:
ORA-01776: cannot modify more than one base table through a join view


SQL>
SQL> REM INSERTION IS NOT POSSIBLE
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Updating values in the table
SQL>
SQL> UPDATE European_Makers SET maker='ford' WHERE country='germany';

4 rows updated.

SQL>
SQL> SELECT * FROM European_Makers WHERE country='germany';

MAKER           FULLNAME                       COUNTRY
--------------- ------------------------------ --------------------
ford            Volkswagen                     germany
ford            BMW                            germany
ford            Daimler Benz                   germany
ford            Opel                           germany

SQL>
SQL>
SQL> REM UPDATION IS POSSIBLE
SQL>
SQL> REM ----------------------------------------------------------------
-
>
SQL> REM Deleting value from table
SQL>
SQL> DELETE FROM European_Makers WHERE maker='fiat';
DELETE FROM European_Makers WHERE maker='fiat'
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.MD_FK) violated - child record
found


SQL>
SQL> REM DELETION IS NOT POSSIBLE
SQL>
SQL> ROLLBACK TO SAVEPOINT S3;

Rollback complete.
```

```
SQL>
SQL> REM CONCLUSION: INSERTION,DELETION IS NOT POSSIBLE BUT UPDATION IS
POSSIBLE
SQL>
SQL> REM
**************************************************************
SQL>
SQL> REM 4. Create a view named Cars_Count which displays total number of
cars manufactured by each country.
SQL>
SQL> REM Creating View and displaying
SQL>
SQL> CREATE OR REPLACE VIEW Cars_Count(car_count,countryname) AS
  2  SELECT COUNT(*),countryname
  3  FROM COUNTRIES co,CAR_NAMES cn,CAR_MAKERS cm,MODEL_DETAILS md
  4  WHERE co.Countryid=cm.country AND cm.Id=md.maker AND
md.model=cn.model
  5  GROUP BY countryname;

View created.

SQL>
SQL> DESC Cars_Count;
 Name
Null?    Type
 ----------------------------------------------------------------------------
----------------------------------------------------------------------------
---------------------------- -------- -------------------------------------
----------------------------------------------------------------------------
--------
 CAR_COUNT
NUMBER
 COUNTRYNAME
VARCHAR2(20)

SQL>
SQL> REM ----------------------------------------------------------------------
-
>
SQL> REM Displaying the rows in the view
SQL>
SQL> SELECT * FROM Cars_Count;

 CAR_COUNT COUNTRYNAME
---------- --------------------
        39 germany
       253 usa
         1 uk
        79 japan
        11 sweden
        14 france
         8 italy

7 rows selected.

SQL>
SQL> SELECT * FROM USER_UPDATABLE_COLUMNS WHERE TABLE_NAME='CARS_COUNT';
```

```
OWNER
TABLE_NAME
------------------------------------------------------------------------------
-------------------------------------------------------- -----------------
------------------------------------------------------------------------------
-------------------------------------
COLUMN_NAME
UPD INS DEL
------------------------------------------------------------------------------
-------------------------------------------------------- --- --- ---
SYSTEM
CARS_COUNT
CAR_COUNT
NO  NO  NO

SYSTEM
CARS_COUNT
COUNTRYNAME
NO  NO  NO


SQL>
SQL> SAVEPOINT S4;

Savepoint created.

SQL>
SQL> REM ------------------------------------------------------------------
-
>
SQL> REM Inserting into view
SQL>
SQL> INSERT INTO Cars_Count VALUES (4,'India');
INSERT INTO Cars_Count VALUES (4,'India')
*
ERROR at line 1:
ORA-01733: virtual column not allowed here


SQL>
SQL> REM INSERTION IS NOT POSSIBLE
SQL>
SQL> REM ------------------------------------------------------------------
-
>
SQL> REM Updating values in the table
SQL>
SQL> UPDATE Cars_Count SET car_count=10 WHERE countryname='usa';
UPDATE Cars_Count SET car_count=10 WHERE countryname='usa'
       *
ERROR at line 1:
ORA-01732: data manipulation operation not legal on this view


SQL>
SQL> REM UPDATION IS NOT POSSIBLE
SQL>
SQL> REM ------------------------------------------------------------------
-
>
```

```
SQL> REM Deleting value from table
SQL>
SQL> DELETE FROM Cars_Count WHERE countryname='germany';
DELETE FROM Cars_Count WHERE countryname='germany'
            *
ERROR at line 1:
ORA-01732: data manipulation operation not legal on this view


SQL>
SQL> REM DELETION IS NOT POSSIBLE
SQL>
SQL> REM CONCLUSION: INSERTION,UPDATION,DELETION IS NOT POSSIBLE
SQL>
SQL> ROLLBACK TO SAVEPOINT S4;

Rollback complete.

SQL>
SQL> REM
**************************************************************
SQL> spool off;
```

```
SQL> @D:\01-College\databaselab\sql_files\ex5a.sql
SQL> REM
**********************************************************************
******************
SQL> set serveroutput on;
SQL> set echo on;
SQL> set linesize 100;
SQL> set pagesize 30;
SQL>
SQL> REM
**********************************************************************
******************
SQL>
SQL> REM 1. Check whether the given model is manufactured by any maker.
If
SQL> REM available, display the maker full name and country else display:
SQL> REM "The given model is not manufactured / Invalid Model"
SQL>
SQL> DECLARE
  2        inp_model MODEL_DETAILS.model%TYPE;
  3        maker CAR_MAKERS.fullname%TYPE;
  4        model MODEL_DETAILS.model%TYPE;
  5        country COUNTRIES.countryname%TYPE;
  6  BEGIN
  7        inp_model := '&inp_model';
  8        SELECT cm.fullname,md.model,co.countryname
  9        INTO maker,model,country
 10        FROM CAR_MAKERS cm,MODEL_DETAILS md,COUNTRIES co
 11        WHERE cm.Id=md.maker AND co.countryid=cm.country AND
inp_model=md.model AND md.model IN(SELECT model FROM CAR_NAMES);
 12        DBMS_OUTPUT.PUT_LINE('----------------------------------------
---------');
 13        DBMS_OUTPUT.PUT_LINE('MODEL '||model||' IS MANUFACTURED');
 14        DBMS_OUTPUT.PUT_LINE('Maker Name: '||maker);
 15        DBMS_OUTPUT.PUT_LINE('Country: '||country);
 16        EXCEPTION
 17             WHEN NO_DATA_FOUND THEN
 18                  DBMS_OUTPUT.PUT_LINE('----------------------------
--------------------');
 19                  DBMS_OUTPUT.PUT_LINE('The given model is not
manufactured / Invalid Model');
 20  END;
 21
 22  /
Enter value for inp_model: mercedes
old   7:   inp_model := '&inp_model';
new   7:   inp_model := 'mercedes';
------------------------------------------------
MODEL mercedes IS MANUFACTURED
Maker Name: Daimler Benz
Country: germany

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL>
SQL> /
Enter value for inp_model: kia
old   7:   inp_model := '&inp_model';
```

```
new   7:   inp_model := 'kia';
-----------------------------------------------
The given model is not manufactured / Invalid Model

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL>
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex5b.sql
SQL>
SQL> REM 2. An user is desired to buy a car with the specific mileage.
Ask the user for
SQL> REM a mileage, and find the car that is equal or closest to the
desired mileage.
SQL> REM Print the car number, model, description and mileage. Also print
the
SQL> REM number of car(s) that is equal or closest to the given mileage.
SQL>
SQL> DECLARE
  2        mileage CAR_DETAILS.mpg%TYPE;
  3        minm NUMBER:= 10000000;
  4        cnt NUMBER := 0;
  5        CURSOR cursor_1 IS SELECT cd.mpg,cd.Id,cn.model,cn.descr FROM
CAR_NAMES cn,CAR_DETAILS cd WHERE cn.Id=cd.Id;
  6        rowcur cursor_1%ROWTYPE;
  7  BEGIN
  8        DBMS_OUTPUT.PUT_LINE(RPAD('CARID',10,' ')||RPAD('MODEL',20,'
')||RPAD('CAR_NAME',30,' ')||RPAD('MILEAGE',10));
  9        DBMS_OUTPUT.PUT_LINE('----------------------------------------
------------------------------------------------------------');
 10        mileage := &mileage;
 11        FOR rowcur IN cursor_1
 12        LOOP
 13             IF(abs(rowcur.mpg-mileage)<minm) THEN
 14                  minm := abs(rowcur.mpg-mileage);
 15             END IF;
 16        END LOOP;
 17        FOR rowcur IN cursor_1
 18        LOOP
 19             IF(abs(rowcur.mpg-mileage)=minm) THEN
 20                  cnt := cnt+1;
 21                  DBMS_OUTPUT.PUT_LINE(RPAD(rowcur.Id,10,'
')||RPAD(rowcur.model,20,' ')||RPAD(rowcur.descr,30,'
')||LPAD(rowcur.mpg,4));
 22             END IF;
 23        END LOOP;
 24        DBMS_OUTPUT.PUT_LINE('----------------------------------------
------------------------------------------------------------');
 25        DBMS_OUTPUT.PUT_LINE(cnt||' car(s) found EQUAL/CLOSEST to
given mileage');
 26  END;
 27
 28  /
Enter value for mileage: 45
old  10:   mileage := &mileage;
new  10:   mileage := 45;
CARID     MODEL               CAR_NAME                       MILEAGE
```

```
--------------------------------------------------------------------------------
--------------------------
337       honda               honda civic 1500 gl           44.6
--------------------------------------------------------------------------------
--------------------------
1 car(s) found EQUAL/CLOSEST to given mileage

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL>
SQL> /
Enter value for mileage: 18
old  10:   mileage := &mileage;
new  10:   mileage := 18;
CARID     MODEL               CAR_NAME                      MILEAGE
--------------------------------------------------------------------------------
--------------------------
1         chevrolet           chevrolet chevelle malibu       18
3         plymouth            plymouth satellite              18
23        amc                 amc hornet                      18
45        amc                 amc matador                     18
53        amc                 amc hornet sportabout (sw)      18
56        ford                ford mustang                    18
84        volvo               volvo 145e (sw)                 18
105       plymouth            plymouth valiant                18
107       amc                 amc hornet                      18
108       ford                ford maverick                   18
115       amc                 amc gremlin                     18
119       mazda               mazda rx3                       18
143       plymouth            plymouth satellite sebring      18
161       chevrolet           chevrolet nova                  18
171       plymouth            plymouth fury                   18
182       ford                ford pinto                      18
208       ford                ford granada ghia               18
--------------------------------------------------------------------------------
--------------------------
17 car(s) found EQUAL/CLOSEST to given mileage

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex5c.sql
SQL> REM 3. For a given country name, display the number of cars
manufactured in
SQL> REM each model by the car makers as shown below. Check for the
availability
SQL> REM of country name.
SQL>
SQL> DECLARE
  2        country COUNTRIES.countryName%TYPE;
  3        country_check varchar(40);
  4        f1 NUMBER := 0;
  5        f2 NUMBER := 0;
  6        tot NUMBER := 0;
  7        CURSOR cursor1 IS
  8        SELECT co.countryname,cm.fullname,cm.maker,COUNT(cn.model) AS
cnt
```

```
  9        FROM CAR_MAKERS cm,COUNTRIES co,MODEL_DETAILS md,CAR_NAMES cn
 10        WHERE cn.model=md.model and md.maker=cm.id and
cm.country=co.countryid
 11        GROUP BY co.countryname,cm.fullname,cm.maker;
 12        cnt_cursor cursor1%ROWTYPE;
 13  BEGIN
 14        country := '&country';
 15        DBMS_OUTPUT.PUT_LINE('-----------------------------------------
------------------------------');
 16        FOR country_check IN(SELECT countryname FROM COUNTRIES)
 17        LOOP
 18             IF country_check.countryname=country THEN
 19                  f1 := 1;
 20             END IF;
 21        END LOOP;
 22        IF f1 =0 THEN
 23             DBMS_OUTPUT.PUT_LINE('COUNTRY NOT IN DATABASE');
 24             goto labelend1;
 25        END IF;
 26        FOR cnt_cursor IN cursor1
 27        LOOP
 28             IF cnt_cursor.countryname=country THEN
 29                  f2 := 1;
 30             END IF;
 31        END LOOP;
 32        IF f2 = 0 THEN
 33             DBMS_OUTPUT.PUT_LINE('The country '||country||' does not
produce any car');
 34             tot := 0;
 35             goto labelend2;
 36        ELSE
 37             DBMS_OUTPUT.PUT_LINE('Country Name:'||'
     '||country);
 38             DBMS_OUTPUT.PUT_LINE(RPAD('Maker Name',20,'
')||RPAD('Model',20,' ')||RPAD('No_of_Cars',30,' '));
 39             DBMS_OUTPUT.PUT_LINE('-----------------------------------
-----------------------------------');
 40             FOR cnt_cursor IN cursor1
 41             LOOP
 42                  IF cnt_cursor.countryname=country THEN
 43
DBMS_OUTPUT.PUT_LINE(RPAD(cnt_cursor.fullname,20,'
')||RPAD(cnt_cursor.maker,20,' ')||RPAD(cnt_cursor.cnt,30,' '));
 44                       tot := tot+cnt_cursor.cnt;
 45                  END IF;
 46             END LOOP;
 47        END IF;
 48        <<labelend2>>
 49             DBMS_OUTPUT.PUT_LINE('-----------------------------------
-----------------------------------');
 50             DBMS_OUTPUT.PUT_LINE(RPAD('TOTAL',40,' ')||tot);
 51        <<labelend1>>
 52             NULL;
 53  END;
 54
 55  /
Enter value for country: korea
old  14:   country := '&country';
new  14:   country := 'korea';
-----------------------------------------------------------------------
```

```
The country korea does not produce any car
------------------------------------------------------------------------
TOTAL                                     0

PL/SQL procedure successfully completed.

SQL> /
Enter value for country: Isreal
old  14:   country := '&country';
new  14:   country := 'Isreal';
------------------------------------------------------------------------
COUNTRY NOT IN DATABASE

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> /
Enter value for country: france
old  14:   country := '&country';
new  14:   country := 'france';
------------------------------------------------------------------------
Country Name:                              france
Maker Name          Model                No_of_Cars
------------------------------------------------------------------------
Peugeaut            peugeaut             8
Citroen             citroen              1
Renault             renault              5
------------------------------------------------------------------------
TOTAL                                     14

PL/SQL procedure successfully completed.

SQL> spool off;
```

```
SQL> @D:\01-College\databaselab\sql_files\ex6a.sql
SQL> set echo on;
SQL> set serveroutput on;
SQL>
SQL> REM:1. Engine capacity or engine displacement is the engine power
which is
SQL> REM measured in cubic inches or cc (cubic centimeter) or liters.
Small cars are
SQL> REM measured using cc (up to 2000 cc). Liters are used for large
cars.
SQL> REM 1 cubic inch = 16.3871 cc
SQL> REM 1 cubic inch = 0.0164 liters
SQL> REM Using the above metric, display the engine capacity in cc or
liters and
SQL> REM categorize into small or large car respectively for the given
car id.
SQL>
SQL>
SQL>
SQL> CREATE OR REPLACE PROCEDURE engine_displacement (carid IN NUMBER)
  2  IS
  3        edispl CAR_DETAILS.edispl%TYPE;
  4  BEGIN
  5        SELECT edispl INTO edispl FROM CAR_DETAILS WHERE Id=carid;
  6        IF edispl*16.3871 < 2000 THEN
  7              DBMS_OUTPUT.PUT_LINE('CAR WITH ID '||carid||' IS A SMALL
CAR WITH ENGINE CAPACITY = '||edispl*16.3871||' CC');
  8        ELSE
  9              DBMS_OUTPUT.PUT_LINE('CAR WITH ID '||carid||' IS A LARGE
CAR WITH ENGINE CAPACITY = '||edispl*0.0164||' LITRES');
 10        END IF;
 11  END;
 12  /

Procedure created.

SQL>
SQL> DECLARE
  2        carid NUMBER;
  3  BEGIN
  4        carid := &carid;
  5        engine_displacement(carid);
  6  END;
  7  /
Enter value for carid: 323
old   4:    carid := &carid;
new   4:    carid := 323;
CAR WITH ID 323 IS A LARGE CAR WITH ENGINE CAPACITY = 2.4764 LITRES

PL/SQL procedure successfully completed.

SQL> /
Enter value for carid: 156
old   4:    carid := &carid;
new   4:    carid := 156;
CAR WITH ID 156 IS A SMALL CAR WITH ENGINE CAPACITY = 1900.9036 CC

PL/SQL procedure successfully completed.

SQL>
```

```
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex6b.sql
SQL> REM 2. Taking a road trip can be the ideal way to see the
countryside. You have planned for a trip to a holiday spot on weekend.
Select the best car
SQL> REM among the given model to reach the spot. The best car is
determined by the lowest fuel consumption cost to the trip. Input the
distance (miles) to
SQL> REM reach the spot and fuel cost ($ / gallons of gas). Fuel
consumption cost = (miles / mpg) x fuel cost.
SQL>
SQL> CREATE OR REPLACE PROCEDURE holiday(cost IN NUMBER,dist IN
NUMBER,inpmodel IN varchar2,destn IN varchar2)
  2  IS
  3        CURSOR modelcur IS SELECT cd.Id,cn.descr,cd.mpg,cn.model
  4        FROM CAR_DETAILS cd,CAR_NAMES cn
  5        WHERE cd.Id=cn.Id AND cn.model=inpmodel;
  6        cur_var modelcur%ROWTYPE;
  7        minm modelcur%ROWTYPE;
  8        minfuel NUMBER:= 10000;
  9        fuelcons NUMBER;
 10
 11  BEGIN
 12
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 13        DBMS_OUTPUT.PUT_LINE('MODEL: '||RPAD(inpmodel,30,'
')||'DESTINATION: '||RPAD(destn,30,' '));
 14        DBMS_OUTPUT.PUT_LINE('DISTANCE(in miles): '||RPAD(dist,17,'
')||'PRICE ($/GALLEONS): '||RPAD(cost,30,' '));
 15
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 16        DBMS_OUTPUT.PUT_LINE(RPAD('Car ID',8,'
')||RPAD('Description',30,' ')||RPAD('MPG',10,' ')||RPAD('Fuel Cost',20,'
'));
 17        FOR cur_var IN modelcur
 18        LOOP
 19              fuelcons := ROUND((dist/cur_var.mpg)*cost,2);
 20              DBMS_OUTPUT.PUT_LINE(RPAD(cur_var.id,8,'
')||RPAD(cur_var.descr,30,' ')||RPAD(cur_var.mpg,10,'
')||RPAD(fuelcons,20,' '));
 21              IF fuelcons < minfuel THEN
 22                    minfuel := fuelcons;
 23                    minm := cur_var;
 24              END IF;
 25        END LOOP;
 26
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 27        DBMS_OUTPUT.PUT_LINE('CAR ID: '||minm.Id||', Name:
'||minm.descr||' is the best car for the trip!');
 28        DBMS_OUTPUT.PUT_LINE('----------------------------------------
--------------');
 29        DBMS_OUTPUT.PUT_LINE('Enjoy your trip with the lowest fuel
consumption cost...');
 30
DBMS_OUTPUT.PUT_LINE('**************************************************
**********');
 31  END;
```

```
 32  /

Procedure created.

SQL>
SQL>
SQL> DECLARE
  2        cost NUMBER;
  3        dist NUMBER;
  4        inpmodel VARCHAR2(20);
  5        destn VARCHAR2(40);
  6  BEGIN
  7        cost := &cost;
  8        dist := &dist;
  9        inpmodel := '&inpmodel';
 10        destn := '&destn';
 11        holiday(cost,dist,inpmodel,destn);
 12  END;
 13  /
Enter value for cost: 2.3
old   7:   cost := &cost;
new   7:   cost := 2.3;
Enter value for dist: 100
old   8:   dist := &dist;
new   8:   dist := 100;
Enter value for inpmodel: opel
old   9:   inpmodel := '&inpmodel';
new   9:   inpmodel := 'opel';
Enter value for destn: San Bay
old  10:   destn := '&destn';
new  10:   destn := 'San Bay';
************************************************************
MODEL: opel                          DESTINATION: San Bay
DISTANCE(in miles): 100              PRICE ($/GALLEONS): 2.3
************************************************************
Car ID  Description                   MPG       Fuel Cost
58      opel 1900                     28        8.21
126     opel manta                    24        9.58
151     opel manta                    26        8.85
191     opel 1900                     25        9.2
************************************************************
CAR ID: 58, Name: opel 1900 is the best car for the trip!
------------------------------------------------------
Enjoy your trip with the lowest fuel consumption cost...
************************************************************

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL>
SQL>
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex6c.sql
SQL> REM 3. The power-to-weight ratio formula for an engine is the power
(hp)
SQL> REM generated by the engine divided by the weight (lbs). This is
commonly
SQL> REM applied to engines and is used as a measurement of performance
of a
```

```
SQL> REM vehicle as whole. Display the car name that has highest, lowest
power-toweight
SQL> REM ratio.
SQL>
SQL> CREATE OR REPLACE PROCEDURE pow_to_weight
  2  IS
  3        CURSOR hp_car IS SELECT cd.horsepower,cd.weight,cn.descr
  4        FROM CAR_DETAILS cd,CAR_NAMES cn
  5        WHERE cd.Id=cn.Id;
  6        hp_cur hp_car%ROWTYPE;
  7        hp_max hp_car%ROWTYPE;
  8        hp_min hp_car%ROWTYPE;
  9        minm NUMBER:= 10000;
 10        maxm NUMBER:= -10000;
 11        ratio NUMBER;
 12  BEGIN
 13        FOR hp_cur IN hp_car
 14        LOOP
 15              ratio := hp_cur.horsepower/hp_cur.weight;
 16              IF ratio < minm THEN
 17                    minm := ratio;
 18                    hp_min := hp_cur;
 19              END IF;
 20              IF ratio > maxm THEN
 21                    maxm := ratio;
 22                    hp_max := hp_cur;
 23              END IF;
 24        END LOOP;
 25        DBMS_OUTPUT.PUT_LINE('HIGHEST POWER TO WEIGHT RATIO =
'||ROUND(hp_max.horsepower/hp_max.weight,2)||' AND CAR NAME IS :
'||hp_max.descr);
 26        DBMS_OUTPUT.PUT_LINE('LOWEST POWER TO WEIGHT RATIO =
'||ROUND(hp_min.horsepower/hp_min.weight,2)||' AND CAR NAME IS :
'||hp_min.descr);
 27  END;
 28  /

Procedure created.

SQL>
SQL> BEGIN
  2        pow_to_weight;
  3  END;
  4  /
HIGHEST POWER TO WEIGHT RATIO = .07 AND CAR NAME IS : buick estate wagon
(sw)
LOWEST POWER TO WEIGHT RATIO = .02 AND CAR NAME IS : vw dasher (diesel)

PL/SQL procedure successfully completed.

SQL>
SQL> @D:\01-College\databaselab\sql_files\ex6d.sql
SQL> REM 4. Develop a stored function which returns the car that exactly
or nearly
SQL> REM matches the given mpg and acceleration of the car. If no car
matches,
SQL> REM then return the car that matches either mpg or acceleration.
SQL>
SQL> CREATE OR REPLACE FUNCTION matchCar(mpg IN NUMBER,acc IN NUMBER,flag
OUT NUMBER) RETURN NUMBER
```

```
  2  IS
  3        carid NUMBER;
  4        CURSOR C1 IS SELECT Id,mpg,accel FROM CAR_DETAILS;
  5        cur_var C1%ROWTYPE;
  6        diffmil NUMBER:= 1000;
  7        diffacc NUMBER:= 1000;
  8  BEGIN
  9        FOR cur_var IN C1
 10        LOOP
 11             IF (abs(mpg-cur_var.mpg)<diffmil) THEN
 12                  diffmil := abs(mpg-cur_var.mpg);
 13             END IF;
 14        END LOOP;
 15        FOR cur_var IN C1
 16        LOOP
 17             IF (abs(acc-cur_var.accel)<diffacc) THEN
 18                  diffacc := abs(acc-cur_var.accel);
 19             END IF;
 20        END LOOP;
 21        FOR cur_var in C1
 22        LOOP
 23             IF (cur_var.mpg=mpg AND cur_var.accel=acc) THEN
 24                  carid := cur_var.Id;
 25                  flag := 1;
 26                  goto end1;
 27             END IF;
 28        END LOOP;
 29        FOR cur_var in C1
 30        LOOP
 31             IF (cur_var.mpg=mpg OR cur_var.accel=acc) THEN
 32                  carid := cur_var.Id;
 33                  flag := 2;
 34                  goto end1;
 35             END IF;
 36        END LOOP;
 37        FOR cur_var in C1
 38        LOOP
 39             IF (abs(acc-cur_var.accel)=diffacc OR abs(mpg-
cur_var.mpg)<diffmil) THEN
 40                  carid := cur_var.Id;
 41                  flag := 3;
 42                  goto end1;
 43             END IF;
 44        END LOOP;
 45        flag := 0;
 46        return carid;
 47        <<end1>>
 48             return carid;
 49  END;
 50  /

Function created.

SQL>
SQL>
SQL> DECLARE
  2        flag NUMBER;
  3        mpg NUMBER;
  4        acc NUMBER;
  5        impg NUMBER;
```

```
  6        iacc NUMBER;
  7        carid NUMBER;
  8        model VARCHAR2(10);
  9        descr VARCHAR2(40);
 10  BEGIN
 11        impg := &impg;
 12        iacc := &iacc;
 13        carid := matchCar(impg,iacc,flag);
 14        IF flag=1 THEN
 15             SELECT cn.model,cn.descr,cd.mpg,cd.accel INTO
model,descr,mpg,acc FROM CAR_NAMES cn,CAR_DETAILS cd WHERE cn.Id=carid
AND cd.Id=carid;
 16             DBMS_OUTPUT.PUT_LINE('MATCH FOR BOTH ACCELERATION AND MPG
FOUND : '||carid);
 17             DBMS_OUTPUT.PUT_LINE('CAR ID : '||carid);
 18             DBMS_OUTPUT.PUT_LINE('CAR MODEL : '||model);
 19             DBMS_OUTPUT.PUT_LINE('CAR NAME : '||descr);
 20             DBMS_OUTPUT.PUT_LINE('MILEAGE : '||mpg);
 21             DBMS_OUTPUT.PUT_LINE('ACCELERATION : '||acc);
 22        END IF;
 23        IF flag=2 THEN
 24             SELECT cn.model,cn.descr,cd.mpg,cd.accel INTO
model,descr,mpg,acc FROM CAR_NAMES cn,CAR_DETAILS cd WHERE cn.Id=carid
AND cd.Id=carid;
 25             DBMS_OUTPUT.PUT_LINE('MATCH FOR EITHER ACCELARATION OR
MPG FOUND: '||carid);
 26             DBMS_OUTPUT.PUT_LINE('CAR ID : '||carid);
 27             DBMS_OUTPUT.PUT_LINE('CAR MODEL : '||model);
 28             DBMS_OUTPUT.PUT_LINE('CAR NAME : '||descr);
 29             DBMS_OUTPUT.PUT_LINE('MILEAGE : '||mpg);
 30             DBMS_OUTPUT.PUT_LINE('ACCELERATION : '||acc);
 31        END IF;
 32        IF flag=3 THEN
 33             SELECT cn.model,cn.descr,cd.mpg,cd.accel INTO
model,descr,mpg,acc FROM CAR_NAMES cn,CAR_DETAILS cd WHERE cn.Id=carid
AND cd.Id=carid;
 34             DBMS_OUTPUT.PUT_LINE('CLOSEST MATCH FOR EITHER
ACCELARATION OR MPG FOUND: '||carid);
 35             DBMS_OUTPUT.PUT_LINE('CAR ID : '||carid);
 36             DBMS_OUTPUT.PUT_LINE('CAR MODEL : '||model);
 37             DBMS_OUTPUT.PUT_LINE('CAR NAME : '||descr);
 38             DBMS_OUTPUT.PUT_LINE('MILEAGE : '||mpg);
 39             DBMS_OUTPUT.PUT_LINE('ACCELERATION : '||acc);
 40        END IF;
 41        IF flag=0 THEN
 42             DBMS_OUTPUT.PUT_LINE('NO MATCH FOR EITHER ACCELARATION OR
MPG FOUND');
 43        END IF;
 44  END;
 45  /
Enter value for impg: 18
old  11:    impg := &impg;
new  11:    impg := 18;
Enter value for iacc: 12
old  12:    iacc := &iacc;
new  12:    iacc := 12;
MATCH FOR BOTH ACCELERATION AND MPG FOUND : 1
CAR ID : 1
CAR MODEL : chevrolet
CAR NAME : chevrolet chevelle malibu
```

```
MILEAGE : 18
ACCELERATION : 12

PL/SQL procedure successfully completed.

SQL> /
Enter value for impg: 19
old  11:    impg := &impg;
new  11:    impg := 19;
Enter value for iacc: 55
old  12:    iacc := &iacc;
new  12:    iacc := 55;
MATCH FOR EITHER ACCELARATION OR MPG FOUND: 41
CAR ID : 41
CAR MODEL : amc
CAR NAME : amc gremlin
MILEAGE : 19
ACCELERATION : 13

PL/SQL procedure successfully completed.

SQL> /
Enter value for impg: 18.8
old  11:    impg := &impg;
new  11:    impg := 18.8;
Enter value for iacc: 12.4
old  12:    iacc := &iacc;
new  12:    iacc := 12.4;
CLOSEST MATCH FOR EITHER ACCELARATION OR MPG FOUND: 30
CAR ID : 30
CAR MODEL : bmw
CAR NAME : bmw 2002
MILEAGE : 26
ACCELERATION : 12.5

PL/SQL procedure successfully completed.

SQL> @D:\01-College\databaselab\sql_files\ex6e.sql
SQL> REM 5. Consider the problem 2. Rewrite it into stored function that
returns the
SQL> REM car ID which consumes minimum fuel cost.
SQL>
SQL> CREATE OR REPLACE FUNCTION min_fuelcost(cost IN NUMBER,dist IN
NUMBER,fuelcons OUT NUMBER,inpmodel IN varchar2) RETURN NUMBER
  2  IS
  3          CURSOR modelcur IS SELECT cd.Id,cn.descr,cd.mpg,cn.model
  4          FROM CAR_DETAILS cd,CAR_NAMES cn
  5          WHERE cd.Id=cn.Id AND cn.model=inpmodel;
  6          cur_var modelcur%ROWTYPE;
  7          minm modelcur%ROWTYPE;
  8          minfuel NUMBER:= 10000;
  9          fuel NUMBER;
 10
 11  BEGIN
 12          FOR cur_var IN modelcur
 13          LOOP
 14                  fuel := ROUND((dist/cur_var.mpg)*cost,2);
 15                  IF fuel < minfuel THEN
 16                          minfuel := fuel;
 17                          minm := cur_var;
```

```
 18                         fuelcons := fuel;
 19                 END IF;
 20         END LOOP;
 21         DBMS_OUTPUT.PUT_LINE(minm.Id);
 22         return minm.Id;
 23  END;
 24  /

Function created.

SQL>
SQL>
SQL> DECLARE
  2         cost NUMBER;
  3         dist NUMBER;
  4         res_min NUMBER;
  5         model VARCHAR2(20);
  6         descr VARCHAR2(50);
  7         destn VARCHAR2(30);
  8         fuelcost NUMBER;
  9         fuelcons NUMBER;
 10         CURSOR modelcur IS SELECT cd.Id,cn.descr,cd.mpg,cn.model
 11         FROM CAR_DETAILS cd,CAR_NAMES cn
 12         WHERE cd.Id=cn.Id;
 13  BEGIN
 14         cost := &cost;
 15         dist := &dist;
 16         model := '&model';
 17         destn := '&destn';
 18         res_min := min_fuelcost(cost,dist,fuelcost,model);
 19
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 20         DBMS_OUTPUT.PUT_LINE('MODEL: '||RPAD(model,30,'
')||'DESTINATION: '||RPAD(destn,30,' '));
 21         DBMS_OUTPUT.PUT_LINE('DISTANCE(in miles): '||RPAD(dist,17,'
')||'PRICE ($/GALLEONS): '||RPAD(cost,30,' '));
 22
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 23         DBMS_OUTPUT.PUT_LINE(RPAD('Car ID',8,'
')||RPAD('Description',30,' ')||RPAD('MPG',10,' ')||RPAD('Fuel Cost',20,'
'));
 24
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 25         FOR cur_var IN modelcur
 26         LOOP
 27                 IF cur_var.model = model THEN
 28                         fuelcons :=
ROUND((dist/cur_var.mpg)*cost,2);
 29                         DBMS_OUTPUT.PUT_LINE(RPAD(cur_var.id,8,'
')||RPAD(cur_var.descr,30,' ')||RPAD(cur_var.mpg,10,'
')||RPAD(fuelcons,20,' '));
 30                 END IF;
 31         END LOOP;
 32         SELECT model,descr INTO model,descr FROM CAR_NAMES WHERE
Id=res_min;
```

```
 33
DBMS_OUTPUT.PUT_LINE('**************************************************
*********');
 34          DBMS_OUTPUT.PUT_LINE('CAR ID WHICH CONSUMES MINUMUM FUEL
COST: '||res_min);
 35          DBMS_OUTPUT.PUT_LINE('CAR MODEL: '||model);
 36          DBMS_OUTPUT.PUT_LINE('CAR NAME: '||descr);
 37          DBMS_OUTPUT.PUT_LINE('FUEL COST: '||fuelcost);
 38          DBMS_OUTPUT.PUT_LINE('-------------------------------------
----------------');
 39  END;
 40  /
Enter value for cost: 2.3
old  14:        cost := &cost;
new  14:        cost := 2.3;
Enter value for dist: 100
old  15:        dist := &dist;
new  15:        dist := 100;
Enter value for model: opel
old  16:        model := '&model';
new  16:        model := 'opel';
Enter value for destn: San Bay
old  17:        destn := '&destn';
new  17:        destn := 'San Bay';
58
************************************************************
MODEL: opel                          DESTINATION: San Bay
DISTANCE(in miles): 100              PRICE ($/GALLEONS): 2.3
************************************************************
Car ID  Description                   MPG       Fuel Cost
************************************************************
58      opel 1900                     28        8.21
126     opel manta                    24        9.58
151     opel manta                    26        8.85
191     opel 1900                     25        9.2
************************************************************
CAR ID WHICH CONSUMES MINUMUM FUEL COST: 58
CAR MODEL: opel
CAR NAME: opel 1900
FUEL COST: 8.21
------------------------------------------------------

PL/SQL procedure successfully completed.

SQL>
SQL>
SQL> spool off;
```

```
SQL> @D:\01-College\databaselab\sql_files\ex7.sql
SQL> REM ********************************************************
SQL> REM              Database Mangement System Lab
SQL> REM              Dept of Computer Science & Engineering
SQL> REM              SSN College of Engineering
SQL> REM ********************************************************
SQL> REM Assignment 8 - PL/SQL TRIGGERS           Date:31-Mar-2021
SQL> REM ---------------------------------------------------------------
----------
>
SQL>
SQL>
SQL> set linesize 200;
SQL> set pagesize 100;
SQL> set echo on;
SQL> set serveroutput on;
SQL>
SQL> REM CUSTOMERS
SQL> REM ---------------
> REM CID, CNAME, AGE, ADDRESS, PHONE
SQL>
SQL> REM DROPPING TABLES
SQL>
SQL> DROP TABLE TRANSACTION;

Table dropped.

SQL> DROP TABLE ACCOUNT;

Table dropped.

SQL> DROP TABLE CUSTOMER;

Table dropped.

SQL>
SQL>
SQL> REM CREATING TABLE CUSTOMERS
SQL>
SQL> CREATE TABLE CUSTOMER(
  2        cid number(3) CONSTRAINT pk_cid PRIMARY KEY,
  3        cname varchar2(20),
  4        age number(2),
  5        address varchar2(50),
  6        phone NUMBER(10)
  7  )
  8  /

Table created.

SQL>
SQL> desc CUSTOMER;
 Name
Null?    Type
 ---------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
---------------------------------------------------
 CID
NOT NULL NUMBER(3)
```

```
 CNAME
VARCHAR2(20)
 AGE
NUMBER(2)
 ADDRESS
VARCHAR2(50)
 PHONE
NUMBER(10)

SQL>
SQL> REM INSERTING VALUES INTO CUSTOMERS
SQL>
SQL> INSERT INTO CUSTOMER VALUES(100,'Adithya',25,'Anna nagar,
Chennai',9843748255);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES(101,'Nikhil Arora',28,'Mogapair West,
Chennai',9345672438);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES(102,'Aradhana',31,'East Tambaram,
Chennai',9523495687);

1 row created.

SQL> INSERT INTO CUSTOMER VALUES(103,'Raghav',34,'Nanganallur,
Chennai',9441245636);

1 row created.

SQL>
SQL> REM DISPLAYING TABLE CUSTOMERS;
SQL>
SQL> SELECT * FROM CUSTOMER;

       CID CNAME                       AGE ADDRESS
PHONE
---------- -------------------- ---------- ------------------------------
-------------------- ----------
       100 Adithya                      25 Anna nagar, Chennai
9843748255
       101 Nikhil Arora                 28 Mogapair West, Chennai
9345672438
       102 Aradhana                     31 East Tambaram, Chennai
9523495687
       103 Raghav                       34 Nanganallur, Chennai
9441245636

SQL>
SQL> REM CREATING TABLE ACCOUNTS
SQL>
SQL> CREATE TABLE ACCOUNT (
  2        ano varchar2(4) CONSTRAINT pk_ano PRIMARY KEY,
  3        atype varchar2(1) CONSTRAINT ch_type CHECK(atype IN
('S','C')),
  4        balance number(7),
  5        cId CONSTRAINT fk_cid REFERENCES CUSTOMER(cid)
  6  )
```

```
  7  /

Table created.

SQL>
SQL> desc ACCOUNT;
 Name
Null?    Type
 ----------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-----------------------------------------------------
 ANO
NOT NULL VARCHAR2(4)
 ATYPE
VARCHAR2(1)
 BALANCE
NUMBER(7)
 CID
NUMBER(3)

SQL>
SQL> REM INSERTING VALUES INTO ACCOUNTS
SQL>
SQL> REM ACCOUNTS
SQL> REM ---------------
> REM ANO, ATYPE, BALANCE, CID
SQL>
SQL> INSERT INTO ACCOUNT VALUES('S103','S',1500,100);

1 row created.

SQL> INSERT INTO ACCOUNT VALUES('C121','C',5000,100);

1 row created.

SQL> INSERT INTO ACCOUNT VALUES('S201','S',45000,101);

1 row created.

SQL> INSERT INTO ACCOUNT VALUES('S223','S',7200,102);

1 row created.

SQL> INSERT INTO ACCOUNT VALUES('C135','C',245000,103);

1 row created.

SQL>
SQL> REM DISPLAYING TABLE ACCOUNTS
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
C121 C       5000        100
S201 S      45000        101
S223 S       7200        102
C135 C     245000        103
```

```
SQL>
SQL> REM CREATING TABLE TRANSACTION
SQL>
SQL> CREATE TABLE TRANSACTION(
  2        tid number CONSTRAINT pk_tid PRIMARY KEY,
  3        aNo varchar2(4) CONSTRAINT fk_ano REFERENCES ACCOUNT(ano),
  4        ttype varchar2(1) CONSTRAINT ch_ttype CHECK(ttype IN
('D','W')),
  5        tdate date,
  6        tamount number
  7  )
  8  /

Table created.

SQL>
SQL> desc TRANSACTION;
 Name
Null?    Type
 ----------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-----------------------------------------------------
 TID
NOT NULL NUMBER
 ANO
VARCHAR2(4)
 TTYPE
VARCHAR2(1)
 TDATE
DATE
 TAMOUNT
NUMBER

SQL>
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex7a.sql
SQL> REM 1. During Withdrawal, check whether the transaction amount is
less than the available
SQL> REM balance. Event : Insert/Update (tamount)
SQL> REM If ttype=Withdrawal and tamount>balance then
SQL> REM raise 'Available Balance is lesser than the Transaction amount'
SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> CREATE OR REPLACE TRIGGER with_check
  2        BEFORE INSERT OR UPDATE ON TRANSACTION
  3        FOR EACH ROW
  4        DECLARE
  5             bal NUMBER;
  6        BEGIN
  7             SELECT balance INTO bal FROM ACCOUNT acc WHERE
acc.ano=:new.ano;
  8             IF :new.tamount>bal AND :new.ttype='W' THEN
  9                  RAISE_APPLICATION_ERROR(-20187,'Available Balance
is lesser than the Transaction amount');
 10             END IF;
 11        END with_check;
 12  /
```

```
Trigger created.

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM INSERTING VALUES INTO TRANSACTION
SQL>
SQL> INSERT INTO TRANSACTION VALUES(1000,'S103','W','01-apr-2021',200);

1 row created.

SQL> INSERT INTO TRANSACTION VALUES(1001,'C121','W','02-apr-2021',1000);

1 row created.

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        200
      1001 C121 W 02-APR-21       1000

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM UPDATING VALUES INTO TRANSACTION
SQL>
SQL> UPDATE TRANSACTION
  2  SET tamount=300 WHERE tid=1000;

1 row updated.

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000

SQL>
```

```
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM INSERTING VALUES TO TRANSACTION TO CHECK TRIGGER
SQL>
SQL> INSERT INTO TRANSACTION VALUES(1002,'S223','W','03-apr-2021',8000);
INSERT INTO TRANSACTION VALUES(1002,'S223','W','03-apr-2021',8000)
            *
ERROR at line 1:
ORA-20187: Available Balance is lesser than the Transaction amount
ORA-06512: at "SYSTEM.WITH_CHECK", line 6
ORA-04088: error during execution of trigger 'SYSTEM.WITH_CHECK'


SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM UPDATING VALUES INTO TRANSACTION
SQL>
SQL> UPDATE TRANSACTION
  2  SET tamount=3000 WHERE tid=1000;
UPDATE TRANSACTION
       *
ERROR at line 1:
ORA-20187: Available Balance is lesser than the Transaction amount
ORA-06512: at "SYSTEM.WITH_CHECK", line 6
ORA-04088: error during execution of trigger 'SYSTEM.WITH_CHECK'


SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000

SQL> @D:\01-College\databaselab\sql_files\ex7b.sql
SQL> REM 2. Implement the following constraint to update the balance of
an account for the
SQL> REM transactions of both the Deposit or Withdrawal type.
SQL> REM Event : Insert/Update (tamount)
SQL> REM If ttype = Deposit then balance = balance + tamount
SQL> REM If ttype = Withdrawal then balance = balance tamount
SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> CREATE OR REPLACE TRIGGER bal_update
  2        BEFORE INSERT OR UPDATE ON TRANSACTION
```

```
  3        FOR EACH ROW
  4        DECLARE
  5              bal NUMBER;
  6        BEGIN
  7              SELECT balance INTO bal FROM ACCOUNT acc WHERE
acc.ano=:new.ano;
  8              IF INSERTING THEN
  9                    IF :new.ttype='W' THEN
 10                          UPDATE ACCOUNT a
 11                          SET a.balance = a.balance-:new.tamount
 12                          WHERE a.ano = :new.ano;
 13                    ELSIF :new.ttype='D' THEN
 14                          UPDATE ACCOUNT a
 15                          SET balance = balance+:new.tamount
 16                          WHERE a.ano = :new.ano;
 17                    END IF;
 18              ELSIF UPDATING THEN
 19                    IF :new.ttype='W' THEN
 20                          UPDATE ACCOUNT a
 21                          SET balance = balance+:old.tamount-
:new.tamount
 22                          WHERE a.ano = :new.ano;
 23                    ELSIF :new.ttype='D' THEN
 24                          UPDATE ACCOUNT a
 25                          SET balance = balance-
:old.tamount+:new.tamount
 26                          WHERE a.ano = :new.ano;
 27                    END IF;
 28              END IF;
 29        END;
 30  /

Trigger created.

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING ACCOUNTS TABLE
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
C121 C       5000        100
S201 S      45000        101
S223 S       7200        102
C135 C     245000        103

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;
```

```
       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM INSERTING VALUES INTO TRANSACTION
SQL>
SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='S223';

ANO  A    BALANCE        CID
---- - ---------- ----------
S223 S       7200        102

SQL>
SQL> INSERT INTO TRANSACTION VALUES(2001,'S223','W','01-apr-2021',1500);

1 row created.

SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='S223';

ANO  A    BALANCE        CID
---- - ---------- ----------
S223 S       5700        102

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='C121';

ANO  A    BALANCE        CID
---- - ---------- ----------
C121 C       5000        100

SQL>
SQL> INSERT INTO TRANSACTION VALUES(2002,'C121','D','02-apr-2021',1000);

1 row created.

SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='C121';

ANO  A    BALANCE        CID
---- - ---------- ----------
C121 C       6000        100

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM UPDATING VALUES INTO TRANSACTION
```

```
SQL>
SQL> SELECT * FROM ACCOUNT a,TRANSACTION t WHERE a.ano=t.ano AND
t.tid=2001 AND t.ttype='W';

ANO  A    BALANCE        CID        TID ANO  T TDATE        TAMOUNT
---- - ---------- ---------- ---------- ---- - --------- ----------
S223 S       5700        102       2001 S223 W 01-APR-21       1500

SQL>
SQL> UPDATE TRANSACTION
  2  SET tamount=1500 WHERE tid=2001 AND ttype='W';

1 row updated.

SQL>
SQL> SELECT * FROM ACCOUNT a,TRANSACTION t WHERE a.ano=t.ano AND
t.tid=2001 AND t.ttype='W';

ANO  A    BALANCE        CID        TID ANO  T TDATE        TAMOUNT
---- - ---------- ---------- ---------- ---- - --------- ----------
S223 S       5700        102       2001 S223 W 01-APR-21       1500

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> SELECT * FROM ACCOUNT a,TRANSACTION t WHERE a.ano=t.ano AND
t.tid=2002 AND t.ttype='D';

ANO  A    BALANCE        CID        TID ANO  T TDATE        TAMOUNT
---- - ---------- ---------- ---------- ---- - --------- ----------
C121 C       6000        100       2002 C121 D 02-APR-21       1000

SQL>
SQL> UPDATE TRANSACTION
  2  SET tamount=2500 WHERE tid=2002 AND ttype='D';

1 row updated.

SQL>
SQL> SELECT * FROM ACCOUNT a,TRANSACTION t WHERE a.ano=t.ano AND
t.tid=2002 AND t.ttype='D';

ANO  A    BALANCE        CID        TID ANO  T TDATE        TAMOUNT
---- - ---------- ---------- ---------- ---- - --------- ----------
C121 C       7500        100       2002 C121 D 02-APR-21       2500

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING ACCOUNT TABLE
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
```

```
C121 C       7500        100
S201 S      45000        101
S223 S       5700        102
C135 C     245000        103

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000
      2001 S223 W 01-APR-21       1500
      2002 C121 D 02-APR-21       2500

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> @D:\01-College\databaselab\sql_files\ex7c.sql
SQL> REM 3. Implement the following constraint for Withdrawal
transactions. Event : Insert
SQL> REM a. A customer can have at most 3 withdrawals per day per
account.
SQL> REM b. The maximum amount of withdrawal for each transaction is
Rs.30000/
SQL>
SQL>
SQL> CREATE OR REPLACE TRIGGER check_withdraw
  2        BEFORE INSERT OR UPDATE ON TRANSACTION
  3        FOR EACH ROW
  4        DECLARE
  5             no_of_withdraw number;
  6        BEGIN
  7             IF :new.tamount>30000 then
  8                  raise_application_error(-20036,'Maximum amount of
withdrawal for each transaction should not exceed Rs.30000');
  9             END IF;
 10             SELECT count(*) into no_of_withdraw FROM transaction t
where t.ano=:new.ano and t.ttype='W' and t.tdate=:new.tdate;
 11             if no_of_withdraw=3 then
 12                  raise_application_error(-20048,'Maximum 3
withdrawal is possible for a customer in a day');
 13             end if;
 14             exception when no_data_found then
 15                  dbms_output.put_line('');
 16        end;
 17  /

Trigger created.

SQL>
```

```
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM DISPLAYING ACCOUNTS TABLE
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
C121 C       7500        100
S201 S      45000        101
S223 S       5700        102
C135 C     245000        103

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000
      2001 S223 W 01-APR-21       1500
      2002 C121 D 02-APR-21       2500

SQL>
SQL> REM
********************************************************************************
************************************
SQL>
SQL> REM INSERTING 3 SAME VALUES INTO TRANSACTION
SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='S223';

ANO  A    BALANCE        CID
---- - ---------- ----------
S223 S       5700        102

SQL>
SQL> INSERT INTO TRANSACTION VALUES(3001,'S223','W','07-apr-2021',1000);

1 row created.

SQL>
SQL> INSERT INTO TRANSACTION VALUES(3002,'S223','W','07-apr-2021',1000);

1 row created.

SQL>
SQL> INSERT INTO TRANSACTION VALUES(3003,'S223','W','07-apr-2021',1000);

1 row created.
```

```
SQL>
SQL> SELECT * FROM ACCOUNT WHERE ano='S223';

ANO  A    BALANCE        CID
---- - ---------- ----------
S223 S       2700        102

SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM INSERTING 4TH WITHDRAWAL FROM SAME ACCOUNT
SQL>
SQL> INSERT INTO TRANSACTION VALUES(3003,'S223','W','07-apr-2021',1000);
INSERT INTO TRANSACTION VALUES(3003,'S223','W','07-apr-2021',1000)
            *
ERROR at line 1:
ORA-20048: Maximum 3 withdrawal is possible for a customer in a day
ORA-06512: at "SYSTEM.CHECK_WITHDRAW", line 9
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_WITHDRAW'


SQL>
SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM DISPLAYING ACCOUNTS TABLE
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
C121 C       7500        100
S201 S      45000        101
S223 S       2700        102
C135 C     245000        103

SQL>
SQL> REM
**************************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000
      2001 S223 W 01-APR-21       1500
      2002 C121 D 02-APR-21       2500
      3001 S223 W 07-APR-21       1000
      3002 S223 W 07-APR-21       1000
      3003 S223 W 07-APR-21       1000
```

```
7 rows selected.

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM UPDATING VALUES INTO TRANSACTION
SQL>
SQL> INSERT INTO TRANSACTION VALUES(3004,'C135','W','02-apr-2021',35000);
INSERT INTO TRANSACTION VALUES(3004,'C135','W','02-apr-2021',35000)
            *
ERROR at line 1:
ORA-20036: Maximum amount of withdrawal for each transaction should not
exceed Rs.30000
ORA-06512: at "SYSTEM.CHECK_WITHDRAW", line 5
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_WITHDRAW'


SQL>
SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING ACCOUNT TABLE
SQL>
SQL> SELECT * FROM ACCOUNT;

ANO  A    BALANCE        CID
---- - ---------- ----------
S103 S       1500        100
C121 C       7500        100
S201 S      45000        101
S223 S       2700        102
C135 C     245000        103

SQL>
SQL> REM
**********************************************************************
************************************
SQL>
SQL> REM DISPLAYING TRANSACTION TABLE
SQL>
SQL> SELECT * FROM TRANSACTION;

       TID ANO  T TDATE        TAMOUNT
---------- ---- - --------- ----------
      1000 S103 W 01-APR-21        300
      1001 C121 W 02-APR-21       1000
      2001 S223 W 01-APR-21       1500
      2002 C121 D 02-APR-21       2500
      3001 S223 W 07-APR-21       1000
      3002 S223 W 07-APR-21       1000
      3003 S223 W 07-APR-21       1000

7 rows selected.

SQL>
SQL>
```

```
SQL>
SQL>
SQL>
SQL> spool off;
```

```
SQL> @D:\01-College\databaselab\sql_files\ex8.sql
SQL> set echo on;
SQL> set linesize 200;
SQL> set pagesize 100;
SQL>
SQL> REM DROPPING TABLES
SQL>
SQL> DROP TABLE ClientRental_Details;

Table dropped.

SQL> DROP TABLE Rental_Details;

Table dropped.

SQL> DROP TABLE Client_Details;

Table dropped.

SQL> DROP TABLE PropertyOwner_Details;

Table dropped.

SQL> DROP TABLE Owner_Details;

Table dropped.

SQL>
SQL> REM
********************************************************************************
********************************************************************
SQL>
SQL> REM CREATING 1NF TABLE
SQL>
SQL> CREATE TABLE ClientRental_Details (
  2        clientNo varchar2(4),
  3        propertyNo varchar2(4),
  4        cName varchar2(15),
  5        pAddress varchar2(60),
  6        rentStart date,
  7        rentFinish date,
  8        rent number(3),
  9        ownerNo varchar2(4),
 10        oName varchar(20),
 11        CONSTRAINT cr_pk PRIMARY KEY(clientNo,propertyNo)
 12  );

Table created.

SQL>
SQL> REM INSERTING INTO 1NF TABLE
SQL>
SQL> INSERT INTO ClientRental_Details VALUES('CR76','PG4','John Kay','6
Lawrence St,Glasgow','01-jul-12','31-aug-13',350,'CO40','Tina Murphy');

1 row created.

SQL> INSERT INTO ClientRental_Details VALUES('CR76','PG16','John Kay','5
Novar Dr,Glasgow','01-sep-13','01-sep-14',450,'CO93','Tony Shaw');
```

```
1 row created.

SQL> INSERT INTO ClientRental_Details VALUES('CR56','PG4','Aline
Stewart','6 Lawrence St,Glasgow','01-sep-11','10-jun-12',350,'CO40','Tina
Murphy');

1 row created.

SQL> INSERT INTO ClientRental_Details VALUES('CR56','PG36','Aline
Stewart','2 Manor Road,Glasgow','10-oct-12','01-dec-13',375,'CO93','Tony
Shaw');

1 row created.

SQL> INSERT INTO ClientRental_Details VALUES('CR56','PG16','Aline
Stewart','5 Novar Dr,Glasgow','01-nov-14','10-aug-15',450,'CO93','Tony
Shaw');

1 row created.

SQL>
SQL> REM DISPLAYING 1NF TABLE
SQL>
SQL> SELECT * FROM ClientRental_Details;

CLIE PROP CNAME           PADDRESS
RENTSTART RENTFINIS       RENT OWNE ONAME
---- ---- --------------- ----------------------------------------------
------------ --------- --------- ---------- ---- --------------------
CR76 PG4  John Kay        6 Lawrence St,Glasgow
01-JUL-12 31-AUG-13        350 CO40 Tina Murphy
CR76 PG16 John Kay        5 Novar Dr,Glasgow
01-SEP-13 01-SEP-14        450 CO93 Tony Shaw
CR56 PG4  Aline Stewart   6 Lawrence St,Glasgow
01-SEP-11 10-JUN-12        350 CO40 Tina Murphy
CR56 PG36 Aline Stewart   2 Manor Road,Glasgow
10-OCT-12 01-DEC-13        375 CO93 Tony Shaw
CR56 PG16 Aline Stewart   5 Novar Dr,Glasgow
01-NOV-14 10-AUG-15        450 CO93 Tony Shaw

SQL>
SQL> REM
**********************************************************************
**********************************************************************
SQL>
SQL> CREATE TABLE Client_Details (
  2        clientNo varchar2(4) CONSTRAINT pk_cn PRIMARY KEY,
  3        cName varchar2(15)
  4  );

Table created.

SQL>
SQL> desc Client_Details;
 Name
Null?    Type
 ----------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-------------------------------------------------
```

```
 CLIENTNO
NOT NULL VARCHAR2(4)
 CNAME
VARCHAR2(15)

SQL>
SQL> REM
********************************************************************************
*****************************************************************
SQL>
SQL> CREATE TABLE Owner_Details (
  2        ownerNo varchar2(4) CONSTRAINT ow_no PRIMARY KEY,
  3        oName varchar(20)
  4  );

Table created.

SQL>
SQL> desc Owner_Details;
 Name
Null?    Type
 ------------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-------------------------------------------------------
 OWNERNO
NOT NULL VARCHAR2(4)
 ONAME
VARCHAR2(20)

SQL>
SQL> REM
********************************************************************************
*****************************************************************
SQL>
SQL> CREATE TABLE PropertyOwner_Details (
  2        propertyNo varchar2(4) CONSTRAINT po_pk PRIMARY KEY,
  3        pAddress varchar2(60),
  4        rent number(3),
  5        ownerNo varchar2(4) CONSTRAINT fk_on REFERENCES
Owner_Details(ownerNo)
  6  );

Table created.

SQL>
SQL> desc PropertyOwner_Details;
 Name
Null?    Type
 ------------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
-------------------------------------------------------
 PROPERTYNO
NOT NULL VARCHAR2(4)
 PADDRESS
VARCHAR2(60)
 RENT
NUMBER(3)
 OWNERNO
VARCHAR2(4)
```

```
SQL>
SQL> REM
********************************************************************************
********************************************************************
SQL>
SQL> CREATE TABLE Rental_Details (
  2        clientNo varchar2(4) CONSTRAINT fk_cn REFERENCES
Client_Details(clientNo),
  3        propertyNo varchar2(4) CONSTRAINT fk_pn REFERENCES
PropertyOwner_Details(propertyNo),
  4        rentStart date,
  5        rentFinish date,
  6        CONSTRAINT rental_pk PRIMARY KEY(clientNo,propertyNo)
  7  );

Table created.

SQL>
SQL> desc Rental_Details;
 Name
Null?    Type
 ----------------------------------------------------------------------------
---------------------------------------- -------- ----------------------
--------------------------------------------------------
 CLIENTNO
NOT NULL VARCHAR2(4)
 PROPERTYNO
NOT NULL VARCHAR2(4)
 RENTSTART
DATE
 RENTFINISH
DATE

SQL>
SQL> REM
********************************************************************************
********************************************************************
SQL>
SQL> REM POPULATING AND DISPLAYING 3NF TABLES
SQL>
SQL> INSERT INTO Client_Details VALUES('CR76','John Kay');

1 row created.

SQL> INSERT INTO Client_Details VALUES('CR56','Aline Stewart');

1 row created.

SQL>
SQL> SELECT * FROM Client_Details;

CLIE CNAME
---- ---------------
CR76 John Kay
CR56 Aline Stewart

SQL>
SQL>
```

```
SQL> REM
**********************************************************************
**********************************************************************
SQL>
SQL> INSERT INTO Owner_Details VALUES('C040','Tina Murphy');

1 row created.

SQL> INSERT INTO Owner_Details VALUES('C093','Tony Shaw');

1 row created.

SQL>
SQL> SELECT * FROM Owner_Details;

OWNE ONAME
---- --------------------
C040 Tina Murphy
C093 Tony Shaw

SQL>
SQL> REM
**********************************************************************
**********************************************************************
SQL>
SQL> INSERT INTO PropertyOwner_Details VALUES('PG4','6 Lawrence
St,Glasgow',350,'C040');

1 row created.

SQL> INSERT INTO PropertyOwner_Details VALUES('PG36','2 Manor
Road,Glasgow',375,'C093');

1 row created.

SQL> INSERT INTO PropertyOwner_Details VALUES('PG16','5 Novar
Dr,Glasgow',450,'C093');

1 row created.

SQL>
SQL> SELECT * FROM PropertyOwner_Details;

PROP PADDRESS
RENT OWNE
---- ------------------------------------------------------------ -------
--- ----
PG4  6 Lawrence St,Glasgow
350 C040
PG36 2 Manor Road,Glasgow
375 C093
PG16 5 Novar Dr,Glasgow
450 C093

SQL>
SQL> REM
**********************************************************************
**********************************************************************
SQL>
```

```
SQL> INSERT INTO Rental_Details VALUES('CR76','PG4','01-jul-12','31-aug-
13');

1 row created.

SQL> INSERT INTO Rental_Details VALUES('CR76','PG16','01-sep-13','01-sep-
14');

1 row created.

SQL> INSERT INTO Rental_Details VALUES('CR56','PG4','01-sep-11','10-jun-
12');

1 row created.

SQL> INSERT INTO Rental_Details VALUES('CR56','PG36','10-oct-12','01-dec-
13');

1 row created.

SQL> INSERT INTO Rental_Details VALUES('CR56','PG16','01-nov-14','10-aug-
15');

1 row created.

SQL>
SQL> SELECT * FROM Rental_Details;

CLIE PROP RENTSTART RENTFINIS
---- ---- --------- ---------
CR76 PG4  01-JUL-12 31-AUG-13
CR76 PG16 01-SEP-13 01-SEP-14
CR56 PG4  01-SEP-11 10-JUN-12
CR56 PG36 10-OCT-12 01-DEC-13
CR56 PG16 01-NOV-14 10-AUG-15

SQL>
SQL> REM
**********************************************************************
**************************************************************
SQL>
SQL> REM PROVING LOSSLESS JOIN PROPERTY USING THE 1NF AND 3NF TABLES
SQL>
SQL> SELECT * FROM ClientRental_Details;

CLIE PROP CNAME           PADDRESS
RENTSTART RENTFINIS       RENT OWNE ONAME
---- ---- --------------- ------------------------------------------------
------------ --------- --------- ---------- ---- --------------------
CR76 PG4  John Kay        6 Lawrence St,Glasgow
01-JUL-12 31-AUG-13        350 CO40 Tina Murphy
CR76 PG16 John Kay        5 Novar Dr,Glasgow
01-SEP-13 01-SEP-14        450 CO93 Tony Shaw
CR56 PG4  Aline Stewart   6 Lawrence St,Glasgow
01-SEP-11 10-JUN-12        350 CO40 Tina Murphy
CR56 PG36 Aline Stewart   2 Manor Road,Glasgow
10-OCT-12 01-DEC-13        375 CO93 Tony Shaw
CR56 PG16 Aline Stewart   5 Novar Dr,Glasgow
01-NOV-14 10-AUG-15        450 CO93 Tony Shaw
```

```
SQL>
SQL> REM JOINING TABLE
SQL>
SQL> SELECT *
  2  FROM Client_Details NATURAL JOIN Rental_Details
  3  NATURAL JOIN PropertyOwner_Details NATURAL JOIN Owner_Details;

OWNE PROP CLIE CNAME           RENTSTART RENTFINIS PADDRESS
RENT ONAME
---- ---- ---- --------------- --------- --------- ----------------------
------------------------------------ ---------- --------------------
C040 PG4  CR76 John Kay        01-JUL-12 31-AUG-13 6 Lawrence St,Glasgow
350 Tina Murphy
C093 PG16 CR76 John Kay        01-SEP-13 01-SEP-14 5 Novar Dr,Glasgow
450 Tony Shaw
C040 PG4  CR56 Aline Stewart   01-SEP-11 10-JUN-12 6 Lawrence St,Glasgow
350 Tina Murphy
C093 PG36 CR56 Aline Stewart   10-OCT-12 01-DEC-13 2 Manor Road,Glasgow
375 Tony Shaw
C093 PG16 CR56 Aline Stewart   01-NOV-14 10-AUG-15 5 Novar Dr,Glasgow
450 Tony Shaw

SQL>
SQL> REM
**********************************************************************
**********************************************************************
SQL>
SQL> REM SINCE THE COLUMN VALUES OF BOTH THE TABLES ARE INTACT AND THERE
HAS BEEN NO LOSS OF DATA IN THE DECOMPOSITION OF TABLES LOSSLESS JOIN
PROPERTY HAS BEEN VERIFIED
SQL>
SQL>
SQL>
SQL>
SQL> spool off;
```