

```
1  drop table patient_table;
2  create table patient_table
3      (Pat_Nbr NUMBER(4,0) PRIMARY KEY,
4      Pat_Name varchar2(25),
5      Pat_Address varchar(25),
6      Pat_City varchar(15),
7      Pat_State char(2),
8      Pat_Zip number(5,0),
9      Pat_Room number(3,0),
10     Pat_Bed number(1,0));
11 insert into patient_table values
12 (1379,'Cribbs, John','2110 Main St.', 'Austin', 'TX', 78711, 101, 1);
13 insert into patient_table values
14 (3249,'Baker, Mary','3547 W. 42nd St.', 'Berkeley', 'CA', 94117, 137, 2);
15 insert into patient_table values
16 (4500,'Garcia, Juan','1533 Telegraph', 'Berkeley', 'CA', 94117, 228, 2);
17 insert into patient_table values
18 (5116,'Harris, Carol','4710 Ave. E', 'Austin', 'TX', 78705, 438, 1);
19 insert into patient_table values
20 (5872,'Zimmer, Elka','7988 Cedar', 'Cleveland', 'OH', 44060, 137, 1);
21 insert into patient_table values
22 (6213,'Rose, David','322 Bridge Ave.', 'Redwood', 'CA', 94065, 100, 1);
23 insert into patient_table values
24 (7459,'Smith, Chris','788 Cummings', 'Cleveland', 'OH', 44066, 438, 3);
25 insert into patient_table values
26 (8031,'Fitch, Sylvia','3380 Fox Ave.', 'Madison', 'WI', 53711, 420, 4);
27 insert into patient_table values
28 (8659,'Hernandez, Juan','8300 Geneva Dr.', 'Austin', 'TX', 78723, 350, 2);
```

```
1  table PATIENT_TABLE dropped.  
2  table PATIENT_TABLE created.  
3  1 rows inserted.  
4  1 rows inserted.  
5  1 rows inserted.  
6  1 rows inserted.  
7  1 rows inserted.  
8  1 rows inserted.  
9  1 rows inserted.  
10 1 rows inserted.  
11 1 rows inserted.
```

```
1  drop table physician_table;
2  create table physician_table
3      (Phys_ID NUMBER(3,0) PRIMARY KEY,
4      Phys_Name varchar2(25),
5      Phys_Phone char(12),
6      Phys_Specialty varchar2(25));
7  insert into physician_table values
8      (101,'Wilcox, Chris','512-329-1848','Eyes, Ears, Throat');
9  insert into physician_table values
10     (102,'Nusca, Jane','512-516-3947','Cardiovascular');
11  insert into physician_table values
12     (103,'Gomez, Juan','512-382-4987','Orthopedics');
13  insert into physician_table values
14     (104,'Li, Jan','512-516-3948','Cardiovascular');
15  insert into physician_table values
16     (105,'Simmons, Alex','512-442-5700','Hemotology');
17  select *
18  from physician_table;
```

```
1   table PHYSICIAN_TABLE dropped.
2   table PHYSICIAN_TABLE created.
3   1 rows inserted.
4   1 rows inserted.
5   1 rows inserted.
6   1 rows inserted.
7   1 rows inserted.
8       PHYS_ID PHYS_NAME                      PHYS_PHONE    PHYS_SPECIALTY
9   -----  -----
10      101 Wilcox, Chris          512-329-1848 Eyes, Ears, Throat
11      102 Nusca, Jane           512-516-3947 Cardiovascular
12      103 Gomez, Juan          512-382-4987 Orthopedics
13      104 Li, Jan              512-516-3948 Cardiovascular
14      105 Simmons, Alex         512-442-5700 Hematology
15
```

```
1  drop table treatment_table;
2  create table treatment_table
3      (Pat_Nbr NUMBER(4,0),
4       Phys_ID NUMBER(3,0),
5       Trt_Procedure varchar2(5),
6       Trt_Date DATE);
7  insert into treatment_table values
8  (3249,101,'13-08','12-FEB-1999');
9  insert into treatment_table values
10 (1379,103,'27-45','25-MAR-1999');
11 insert into treatment_table values
12 (3249,103,'88-20','22-JAN-1999');
13 insert into treatment_table values
14 (5116,104,'52-14','03-APR-1999');
15 insert into treatment_table values
16 (4500,101,'13-08','04-FEB-1999');
17 insert into treatment_table values
18 (8031,102,'52-14','15-MAR-2000');
19 insert into treatment_table values
20 (5116,104,'52-14','05-FEB-2001');
21 insert into treatment_table values
22 (5872,105,'60-00','13-FEB-2000');
23 insert into treatment_table values
24 (3249,103,'88-20','24-JAN-2000');
25 insert into treatment_table values
26 (8659,104,'60-00','08-APR-2001');
```

```
1  table TREATMENT_TABLE dropped.  
2  table TREATMENT_TABLE created.  
3  1 rows inserted.  
4  1 rows inserted.  
5  1 rows inserted.  
6  1 rows inserted.  
7  1 rows inserted.  
8  1 rows inserted.  
9  1 rows inserted.  
10 1 rows inserted.  
11 1 rows inserted.  
12 1 rows inserted.
```

```
1  drop table procedure_table;
2  create table procedure_table
3      (Pro_Nbr CHAR(5) PRIMARY KEY,
4       Pro_Desc varchar2(30),
5       Pro_Charge NUMBER(6,2));
6  insert into procedure_table values
7  ('13-08','Throat culture',15.00);
8  insert into procedure_table values
9  ('27-45','X-Ray',62.00);
10 insert into procedure_table values
11 ('52-14','Cardiogram',135.00);
12 insert into procedure_table values
13 ('60-00','Blood Analysis',58.00);
14 insert into procedure_table values
15 ('88-20','MRI',800.00);
```

```
1  Error starting at line 1 in command:  
2  drop table procedure_table  
3  Error report:  
4  SQL Error: ORA-00942: table or view does not exist  
5  00942. 00000 - "table or view does not exist"  
6  *Cause:  
7  *Action:  
8  table PROCEDURE_TABLE created.  
9  1 rows inserted.  
10 1 rows inserted.  
11 1 rows inserted.  
12 1 rows inserted.  
13 1 rows inserted.
```

```
1  create or replace procedure print_purchase_history
2      (p_lowerCustomerID_num IN customer.cust_id%type,
3       p_upperCustomerID_num IN customer.cust_id%type)
4   IS
5       CURSOR cust_po_cursor IS
6           select customer.cust_id, purchase_order.order_id, purchase_order \
7               .purchase_date
8                   from customer left join purchase_order
9                     on customer.cust_id = purchase_order.cust_id
10                    where customer.cust_id between p_lowerCustomerID_num and p_upperCustomerID_num
11                        order by cust_id;
12   Type cust_po_type is record
13     (custID customer.cust_id%type,
14      orderID purchase_order.order_id%type,
15      purchDate purchase_order.purchase_date%type);
16   cust_po_record cust_po_type;
17   lv_lastCustId_num customer.cust_id%type := 0;
18   invalidSequence_ex Exception;
19   no_entry_found_ex Exception;
20   BEGIN
21     if p_lowerCustomerID_num >= p_upperCustomerID_num then
22       raise invalidSequence_ex;
23     end if;
24     open cust_po_cursor;
25     Loop
26       fetch cust_po_cursor into cust_po_record;
27
28       --If cursor has no entries than entire range of customer IDs does not have entries in table
29       if cust_po_record.custID is null then
30         raise no_entry_found_ex;
31       end if;
32
33       --If first customer ID fetched from cursor is greater than start of range
34       --then customer ID at start of range does not have an entry in table
35       if (lv_lastCustID_num = 0) and (cust_po_record.custID > p_lowerCustomerID_num) then
36         raise no_entry_found_ex;
37       end if;
```

```
38      --if end of cursor has been reached but we have not yet reached \
the upper limit of
39      --range of customer IDs then upper limit ID does not have entry \
in table
40      if (cust_po_cursor%NOTFOUND) and (lv_lastCustID_num < p_upperCus \
tomerID_num) then
41          raise no_entry_found_ex;
42      end if;
43      exit when cust_po_cursor%NOTFOUND;
44      if cust_po_record.custID <> lv_lastCustId_num then
45          if cust_po_record.orderID is null then
46              DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Customer with id ' || cu \
st_po_record.custID || ' does not have a purchase order.');
47          else
48              DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Customer with id ' || cu \
st_po_record.custID || ' has a purchase order.');
49          end if;
50      end if;
51      lv_lastCustId_num := cust_po_record.custID;
52  end loop;
53  close cust_po_cursor;
54 EXCEPTION
55  when NO_DATA_FOUND then
56      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Error: No entry exists for 1 or \
more customer IDs in given range.');
57  when no_entry_found_ex then
58      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Error: No entry exists for 1 or \
more customer IDs in given range.');
59  when invalidSequence_ex then
60      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Invalid range given. First numbe \
r must be less than second number.');
61  when others then
62      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'An error has occurred and techni \
cal support has been notified.');
63  end print_purchase_history;
```

```
1 PROCEDURE PRINT_PURCHASE_HISTORY compiled
```

```
1  <<test_print_purchase_history>>
2  declare
3      lv_lowTestCustID_num customer.cust_id%type;
4      lv_highTestCustID_num customer.cust_id%type;
5  begin
6      lv_lowTestCustID_num := 90001;
7      lv_highTestCustID_num := 90008;
8      DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to ' \
|| lv_highTestCustID_num);
9      print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
10
11     lv_lowTestCustID_num := 90003;
12     lv_highTestCustID_num := 90007;
13     DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to ' \
|| lv_highTestCustID_num);
14     print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
15
16     --entire range has no entries in table
17     lv_lowTestCustID_num := 90009;
18     lv_highTestCustID_num := 90010;
19     DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to ' \
|| lv_highTestCustID_num);
20     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Confirm error when entire range has \
no entries.');
21     print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
22
23     --first number is greater than second number
24     lv_lowTestCustID_num := 90005;
25     lv_highTestCustID_num := 90004;
26     DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to ' \
|| lv_highTestCustID_num);
27     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Confirm error when first number high \
er.');
28     print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
29
30     --start of range does not have entry in table
31     lv_lowTestCustID_num := 90000;
32     lv_highTestCustID_num := 90004;
33     DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to ' \
|| lv_highTestCustID_num);
34     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Confirm error when start of range ha \
s no table entry.');
35     print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
```

```
36
37      --end of range has no entries in table
38      lv_lowTestCustID_num := 90006;
39      lv_highTestCustID_num := 90010;
40      DBMS_OUTPUT.PUT_LINE('For range ' || lv_lowTestCustID_num || ' to '  \
|| lv_highTestCustID_num);
41      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Confirm error when end of range has  \
no table entry.');
42      print_purchase_history(lv_lowTestCustID_num, lv_highTestCustID_num);
43
44  end test_print_purchase_history;
```

```
1 anonymous block completed
```

```
1  For range 90001 to 90008
2    Customer with id 90001 has a purchase order.
3    Customer with id 90002 does not have a purchase order.
4    Customer with id 90003 does not have a purchase order.
5    Customer with id 90004 has a purchase order.
6    Customer with id 90005 has a purchase order.
7    Customer with id 90006 does not have a purchase order.
8    Customer with id 90007 has a purchase order.
9    Customer with id 90008 has a purchase order.
10 For range 90003 to 90007
11   Customer with id 90003 does not have a purchase order.
12   Customer with id 90004 has a purchase order.
13   Customer with id 90005 has a purchase order.
14   Customer with id 90006 does not have a purchase order.
15   Customer with id 90007 has a purchase order.
16 For range 90009 to 90010
17   Confirm error when entire range has no entries.
18   Error: No entry exists for 1 or more customer IDs in given range.
19 For range 90005 to 90004
20   Confirm error when first number higher.
21   Invalid range given. First number must be less than second number.
22 For range 90000 to 90004
23   Confirm error when start of range has no table entry.
24   Error: No entry exists for 1 or more customer IDs in given range.
25 For range 90006 to 90010
26   Confirm error when end of range has no table entry.
27   Customer with id 90006 does not have a purchase order.
28   Customer with id 90007 has a purchase order.
29   Customer with id 90008 has a purchase order.
30   Error: No entry exists for 1 or more customer IDs in given range.
31
```

```
1  create or replace package Hospital
2    IS
3      type patTreatmentRecord is record(  

4        Pat_Nbr Treatment_Table.Pat_Nbr%type,  

5        Trt_Procedure Treatment_Table.Trt_Procedure%type,  

6        Phys_ID Physician_Table.Phys_ID%type,  

7        Phys_Name Physician_Table.Phys_Name%type,  

8        Phys_Specialty Physician_Table.Phys_Specialty%type);  

9      type t_patTrt is table of patTreatmentRecord  

10     index by binary_integer;  

11   e_DupPhysFound exception;  

12   Procedure BuildPatTbl  

13     (t_patientTable OUT t_patTrt,  

14      no_of_rows IN OUT number);  

15   Function FindPatient  

16     (patientNo IN Patient_Table.Pat_Nbr%type)  

17     RETURN boolean;  

18   Function FindPatient  

19     (patientName IN Patient_Table.Pat_Name%type)  

20     RETURN boolean;  

21   Procedure NewPhys  

22     (PhysID IN Physician_Table.Phys_ID%type,  

23      PhysName IN Physician_Table.Phys_Name%type,  

24      PhysPhone IN Physician_Table.Phys_Phone%type,  

25      PhysSpecialty IN Physician_Table.Phys_Specialty%type);  

26 end Hospital;
```

```
1 PACKAGE HOSPITAL compiled
```

```
1  create or replace package body Hospital
2  IS
3  Procedure BuildPatTbl
4      (t_patientTable OUT t_patTrt,
5       no_of_rows IN OUT number)
6  IS
7  BEGIN
8      select Treatment_Table.Pat_Nbr, Treatment_Table.Trt_Procedure,
9          Physician_Table.Phys_ID, Physician_Table.Phys_Name,
10         Physician_Table.Phys_Specialty
11     bulk collect into t_patientTable
12     from Treatment_Table, Physician_Table
13     where Treatment_Table.Phys_ID = Physician_Table.Phys_ID
14     order by Treatment_Table.Pat_Nbr;
15     no_of_rows := t_patientTable.COUNT;
16 END BuildPatTbl;
17
18 Function FindPatient
19     (patientNo IN Patient_Table.Pat_Nbr%type)
20     RETURN boolean
21 IS
22     lv_patientCount_num number(1) :=0;
23 BEGIN
24     select COUNT(Pat_Nbr)
25         into lv_patientCount_num
26         from Patient_Table
27         where Pat_Nbr = patientNo;
28     if lv_patientCount_num > 0 then
29         return true;
30     else
31         return false;
32     end if;
33 end FindPatient;
34
35 Function FindPatient
36     (patientName IN Patient_Table.Pat_Name%type)
37     RETURN boolean
38 IS
39     lv_patientCount_num number(1) := 0;
40 BEGIN
41     select COUNT(Pat_Name)
42         into lv_patientCount_num
43         from Patient_Table
```

```
44      where Pat_Name = patientName;
45      if lv_patientCount_num > 0 then
46          return true;
47      else
48          return false;
49      end if;
50  end FindPatient;
51
52 Procedure NewPhys
53     (PhysID IN Physician_Table.Phys_ID%type,
54      PhysName IN Physician_Table.Phys_Name%type,
55      PhysPhone IN Physician_Table.Phys_Phone%type,
56      PhysSpecialty IN Physician_Table.Phys_Specialty%type)
57 IS
58     lv_physicianCount_num number(1) := 0;
59     physTableRow Physician_Table%rowtype;
60 BEGIN
61     select COUNT(Phys_ID)
62         into lv_physicianCount_num
63         from Physician_Table
64         where Physician_Table.Phys_ID = PhysID;
65     if lv_physicianCount_num > 0 then
66         raise e_DupPhysFound;
67     end if;
68     physTableRow.Phys_ID := PhysID;
69     physTableRow.Phys_Name := PhysName;
70     physTableRow.Phys_Phone := PhysPhone;
71     physTableRow.Phys_Specialty := PhysSpecialty;
72     insert into Physician_Table
73         Values physTableRow;
74     commit;
75  end NewPhys;
76
77 end Hospital;
```

```
1 PACKAGE BODY HOSPITAL compiled
```

```
1  declare
2      lv_patientNum_num  Patient_Table.Pat_Nbr%type;
3      lv_patientName_txt  Patient_Table.Pat_Name%type;
4      physicianTableRow  Physician_Table%rowtype;
5      patientTreatTable  Hospital.t_patTrt;
6      lv_treatmentCount_num  number(2) := 0;
7
8  begin
9      DBMS_OUTPUT.PUT_LINE('-----');
10     DBMS_OUTPUT.PUT_LINE('Test FindPatient Function');
11     DBMS_OUTPUT.PUT_LINE('-----');
12     DBMS_OUTPUT.PUT_LINE(' ');
13     lv_patientNum_num := 4500;
14     DBMS_OUTPUT.PUT_LINE('Verifying Patient ' || lv_patientNum_num || ' \
15 :');
16     if Hospital.FindPatient(lv_patientNum_num) = true then
17         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient exists.');
18     else
19         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient does not exist.');
20     end if;
21
22     lv_patientName_txt := 'Baker, Mary';
23     DBMS_OUTPUT.PUT_LINE('Verifying Patient ' || lv_patientName_txt || ' \
24 :');
25     if Hospital.FindPatient(lv_patientName_txt) = true then
26         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient exists.');
27     else
28         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient does not exist.');
29     end if;
30
31     lv_patientNum_num := 4501;
32     DBMS_OUTPUT.PUT_LINE('Verifying Patient ' || lv_patientNum_num || ' \
33 :');
34     if Hospital.FindPatient(lv_patientNum_num) = true then
35         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient exists.');
36     else
37         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'This patient does not exist.');
38     end if;
39
40     DBMS_OUTPUT.PUT_LINE(' ');
41     DBMS_OUTPUT.PUT_LINE('-----');
42     DBMS_OUTPUT.PUT_LINE('Test BuildPatTbl Procedure');
43     DBMS_OUTPUT.PUT_LINE('-----');
```

```
41      DBMS_OUTPUT.PUT_LINE(' ');
42
43      Hospital.BuildPatTbl(patientTreatTable, lv_treatmentCount_num);
44      DBMS_OUTPUT.PUT_LINE('Patient treatment table with ' || lv_treatment \
tCount_num || ' rows created.');
45      FOR currentRow IN 1..patientTreatTable.COUNT LOOP
46          DBMS_OUTPUT.PUT_LINE('Row ' || currentRow);
47          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient Number: ' || patientTreat \
Table(currentRow).Pat_Nbr);
48          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment Procedure: ' || patient \
TreatTable(currentRow).Trt_Procedure);
49          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || patientTreatTa \
ble(currentRow).Phys_ID);
50          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician Name: ' || patientTreat \
Table(currentRow).Phys_Name);
51          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician Specialty: ' || patient \
TreatTable(currentRow).Phys_Specialty);
52      end loop;
53
54      DBMS_OUTPUT.PUT_LINE(' ');
55      DBMS_OUTPUT.PUT_LINE('-----');
56      DBMS_OUTPUT.PUT_LINE('Test NewPhys Procedure');
57      DBMS_OUTPUT.PUT_LINE('-----');
58      DBMS_OUTPUT.PUT_LINE(' ');
59
60      delete from Physician_Table
61      where Phys_ID = 106;
62      commit;
63
64      physicianTableRow.Phys_ID := 106;
65      physicianTableRow.Phys_Name := 'Swatch, Becky';
66      physicianTableRow.Phys_Phone := '978-985-1234';
67      physicianTableRow.Phys_Specialty := 'Hemotology';
68
69      Hospital.NewPhys(physicianTableRow.Phys_ID, physicianTableRow.Phys_ \
Name,
70                      physicianTableRow.Phys_Phone, physicianTableRow.Phys_Specialty);
71
72      DBMS_OUTPUT.PUT_LINE('Read values into physicianTableRow to clear c \
urrent values.');
73      select *
74          into physicianTableRow
75          from Physician_Table
```

```
76      where Phys_ID = 101;
77      DBMS_OUTPUT.PUT_LINE('Readback of older physician entry.');
78      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || physicianTableRow \
w.Phys_ID);
79      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Name: ' || physicianTableRow.Phys_N \
ame);
80      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Phone: ' || physicianTableRow.Phys_ \
Phone);
81      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Speciality: ' || physicianTableRow. \
Phys_Specialty);
82
83      DBMS_OUTPUT.PUT_LINE('Readback new physician added to confirm it wa \
s written to table.');
84      select *
85          into physicianTableRow
86          from Physician_Table
87          where Phys_ID = 106;
88      DBMS_OUTPUT.PUT_LINE('Readback of new physician entry.');
89      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || physicianTableRow \
w.Phys_ID);
90      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Name: ' || physicianTableRow.Phys_N \
ame);
91      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Phone: ' || physicianTableRow.Phys_ \
Phone);
92      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Speciality: ' || physicianTableRow. \
Phys_Specialty);
93
94
95      physicianTableRow.Phys_ID := 105;
96      physicianTableRow.Phys_Name := 'Simmons, Alex';
97      physicianTableRow.Phys_Phone := '512-442-5700';
98      physicianTableRow.Phys_Specialty := 'Hematology';
99
100     DBMS_OUTPUT.PUT_LINE('About to create duplicate entry.');
101     Hospital.NewPhys(physicianTableRow.Phys_ID, physicianTableRow.Phys_ \
Name,
102                     physicianTableRow.Phys_Phone, physicianTableRow.Phys_Specialty);
103
104 Exception
105     when HOSPITAL.e_DupPhysFound then
106         DBMS_OUTPUT.PUT_LINE('Error: attempt to enter physician who alrea \
dy exists in Database.');
```

108 **end;**

```
1 anonymous block completed
```

```
1 -----  
2 Test FindPatient Function  
3 -----  
4  
5 Verifying Patient 4500:  
6     This patient exists.  
7 Verifying Patient Baker, Mary:  
8     This patient exists.  
9 Verifying Patient 4501:  
10    This patient does not exist.  
11  
12 -----  
13 Test BuildPatTbl Procedure  
14 -----  
15  
16 Patient treatment table with 11 rows created.  
17 Row 1  
18     Patient Number: 1379  
19     Treatment Procedure: 27-45  
20     Physician ID: 103  
21     Physician Name: Gomez, Juan  
22     Physician Specialty: Orthopedics  
23 Row 2  
24     Patient Number: 2013  
25     Treatment Procedure: 13-08  
26     Physician ID: 101  
27     Physician Name: Wilcox, Chris  
28     Physician Specialty: Eyes, Ears, Throat  
29 Row 3  
30     Patient Number: 3249  
31     Treatment Procedure: 13-08  
32     Physician ID: 101  
33     Physician Name: Wilcox, Chris  
34     Physician Specialty: Eyes, Ears, Throat  
35 Row 4  
36     Patient Number: 3249  
37     Treatment Procedure: 88-20  
38     Physician ID: 103  
39     Physician Name: Gomez, Juan  
40     Physician Specialty: Orthopedics  
41 Row 5  
42     Patient Number: 3249  
43     Treatment Procedure: 88-20
```

```
44 Physician ID: 103
45 Physician Name: Gomez, Juan
46 Physician Specialty: Orthopedics
47 Row 6
48 Patient Number: 4500
49 Treatment Procedure: 13-08
50 Physician ID: 101
51 Physician Name: Wilcox, Chris
52 Physician Specialty: Eyes, Ears, Throat
53 Row 7
54 Patient Number: 5116
55 Treatment Procedure: 52-14
56 Physician ID: 104
57 Physician Name: Li, Jan
58 Physician Specialty: Cardiovascular
59 Row 8
60 Patient Number: 5116
61 Treatment Procedure: 52-14
62 Physician ID: 104
63 Physician Name: Li, Jan
64 Physician Specialty: Cardiovascular
65 Row 9
66 Patient Number: 5872
67 Treatment Procedure: 60-00
68 Physician ID: 105
69 Physician Name: Simmons, Alex
70 Physician Specialty: Hematology
71 Row 10
72 Patient Number: 8031
73 Treatment Procedure: 52-14
74 Physician ID: 102
75 Physician Name: Nusca, Jane
76 Physician Specialty: Cardiovascular
77 Row 11
78 Patient Number: 8659
79 Treatment Procedure: 60-00
80 Physician ID: 104
81 Physician Name: Li, Jan
82 Physician Specialty: Cardiovascular
83
84 -----
85 Test NewPhys Procedure
86 -----
```

```
87
88  Read values into physicianTableRow to clear current values.
89  Readback of older physician entry.
90      Physician ID: 101
91      Name: Wilcox, Chris
92      Phone: 512-329-1848
93      Speciality: Eyes, Ears, Throat
94  Readback new physician added to confirm it was written to table.
95  Readback of new physician entry.
96      Physician ID: 106
97      Name: Swatch, Becky
98      Phone: 978-985-1234
99      Speciality: Hematology
100 About to create duplicate entry.
101 Error: attempt to enter physician who already exists in Database.
102
```

```
1    <<modify_hospital_tables>>
2    Declare
3        lv_testPhysID_num  Physician_Table.Phys_ID%type;
4        physTableRow  Physician_Table%rowtype;
5        lv_physicianPhone_txt varchar2(12);
6    begin
7
8        lv_testPhysID_num := 203;
9
10       physTableRow.Phys_ID := lv_testPhysID_num;
11       physTableRow.Phys_Name := 'Mandy Astra';
12       physTableRow.Phys_Phone := '202-571-1234';
13       physTableRow.Phys_Specialty := 'Hematology';
14
15       lv_physicianPhone_txt := '512-310-1234';
16
17       insert into Physician_Table
18           Values physTableRow;
19
20       select Phys_ID, Phys_Name, Phys_Phone, Phys_Specialty
21           into physTableRow
22           from Physician_Table
23           where Phys_ID = lv_testPhysID_num;
24
25       DBMS_Output.Put_Line('Read back from table to confirm insert into  \
26 Physician Table for Lab 4.4');
27       DBMS_Output.Put_Line(CHR(9) || 'Phys ID: ' || physTableRow.Phys_ID \
28 );
28       DBMS_Output.Put_Line(CHR(9) || 'Phys Name: ' || physTableRow.Phys_ \
29 Name);
30       DBMS_Output.Put_Line(CHR(9) || 'Phys Phone: ' || physTableRow.Phys \
31 _Phone);
32       DBMS_Output.Put_Line(CHR(9) || 'Phys Specialty: ' || physTableRow. \
33 Phys_Specialty);
34
35       update Physician_Table
36           set Phys_Phone = lv_physicianPhone_txt
37           where Phys_ID = lv_testPhysID_num;
38
39
40       select Phys_ID, Phys_Name, Phys_Phone, Phys_Specialty
41           into physTableRow
42           from Physician_Table
```

```
39      where Phys_ID = lv_testPhysID_num;
40
41      DBMS_Output.Put_Line('Read back to confirm phone update of Physician \
an Table for Lab 4.4');
42      DBMS_Output.Put_Line(CHR(9) || 'Phys ID: ' || physTableRow.Phys_ID \
);
43      DBMS_Output.Put_Line(CHR(9) || 'Phys Name: ' || physTableRow.Phys_ \
Name);
44      DBMS_Output.Put_Line(CHR(9) || 'Phys Phone: ' || physTableRow.Phys_ \
_Phase);
45      DBMS_Output.Put_Line(CHR(9) || 'Phys Specialty: ' || physTableRow. \
Phys_Specialty);
46
47      delete from Physician_Table
48      where Phys_ID = lv_testPhysID_num;
49
50 end modify_hospital_tables;
```

```
1 anonymous block completed
```

```
1   Read back from table to confirm insert into Physician Table for Lab 4. \
4
2   Phys ID: 203
3   Phys Name: Mandy Astra
4   Phys Phone: 202-571-1234
5   Phys Specialty: Hematology
6   Read back to confirm phone update of Physician Table for Lab 4.4
7   Phys ID: 203
8   Phys Name: Mandy Astra
9   Phys Phone: 512-310-1234
10  Phys Specialty: Hematology
11
```

```
1  create or replace trigger check_treatment_date
2    before insert or update
3    of Trt_date on Treatment_Table
4    for each row
5    -- when ((NEW.Trt_date > SYSDATE) or (MONTHS_BETWEEN(SYSDATE, NEW.Trt \
6      _date) > 3))
6    begin
7      if ((:NEW.Trt_date > SYSDATE) or (MONTHS_BETWEEN(SYSDATE, :NEW.Trt_d \
ate) > 3)) then
8        DBMS_OUTPUT.PUT_LINE('Trigger will generate error.');
9        RAISE_APPLICATION_ERROR(-20100, 'Invalid treatment date.');
10      else
11        DBMS_OUTPUT.PUT_LINE('Trigger gives no error.');
12      end if;
13    end;
```

```
1    TRIGGER CHECK_TREATMENT_DATE compiled
```

```
1  declare
2      treatmentTableRow Treatment_Table%rowtype;
3  begin
4      treatmentTableRow.Pat_Nbr := 2013;
5      treatmentTableRow.Phys_ID := 101;
6      treatmentTableRow.Trn_Procedure := '13-08';
7      treatmentTableRow.Trn_Date := '04-Nov-2013';
8
9      delete from Treatment_Table
10     where Pat_Nbr = 2013;
11
12      commit;
13
14      insert into Treatment_Table
15         Values treatmentTableRow;
16
17      commit;
18
19      select *
20          into treatmentTableRow
21      from Treatment_Table
22      where Pat_Nbr = 2013;
23
24      DBMS_OUTPUT.PUT_LINE('Read back after insert with valid date.');
25      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.Pat_Nbr);
26      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow.Phys_ID);
27      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Trn_Procedure);
28      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow.Trn_Date);
29
30      update Treatment_Table
31         set Trn_Date = '30-Nov-2013'
32       where Pat_Nbr = 2013;
33
34      commit;
35
36      select *
37          into treatmentTableRow
38      from Treatment_Table
39      where Pat_Nbr = 2013;
40
41      DBMS_OUTPUT.PUT_LINE('Read back after change to valid date.');
42      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.Pat_Nbr);
43      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow.Phys_ID);
44      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Trn_Procedure);
```

```
        t_Procedure);
38      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableR \
ow.Trt_Date);
39
40    update Treatment_Table
41      set Trt_Date = '04-Dec-2013'
42      where Pat_Nbr = 2013;
43    select *
44      into treatmentTableRow
45      from Treatment_Table
46      where Pat_Nbr = 2013;
47    DBMS_OUTPUT.PUT_LINE('Read back after change to valid date.');
48    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \
at_Nbr);
49    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow \
.Phys_ID);
50    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \
t_Procedure);
51    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableR \
ow.Trt_Date);
52  end;
```

```
1 anonymous block completed
```

```
1 Trigger gives no error.  
2 Read back after insert with valid date.  
3 Patient No: 2013  
4 Physician ID: 101  
5 Procedure: 13-08  
6 Treatment date: 04-NOV-13  
7 Trigger gives no error.  
8 Read back after change to valid date.  
9 Patient No: 2013  
10 Physician ID: 101  
11 Procedure: 13-08  
12 Treatment date: 30-NOV-13  
13 Trigger gives no error.  
14 Read back after change to valid date.  
15 Patient No: 2013  
16 Physician ID: 101  
17 Procedure: 13-08  
18 Treatment date: 04-DEC-13  
19
```

```
1  declare
2      treatmentTableRow Treatment_Table%rowtype;
3  begin
4      treatmentTableRow.Pat_Nbr := 2013;
5      treatmentTableRow.Phys_ID := 101;
6      treatmentTableRow.Trn_Procedure := '13-08';
7      treatmentTableRow.Trn_Date := '04-Nov-2013';
8
9      delete from Treatment_Table
10     where Pat_Nbr = 2013;
11
12     commit;
13
14     DBMS_OUTPUT.PUT_LINE('Insert with valid date.');
15     insert into Treatment_Table
16         Values treatmentTableRow;
17
18     commit;
19     select *
20         into treatmentTableRow
21         from Treatment_Table
22         where Pat_Nbr = 2013;
23     DBMS_OUTPUT.PUT_LINE('Read back after insert with valid date.');
24     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \ 
at_Nbr);
25     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow \ 
.Phys_ID);
26     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \ 
n_Procedure);
27     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow \ 
.Trn_Date);
28
29     DBMS_OUTPUT.PUT_LINE('Change to valid date.');
30     update Treatment_Table
31         set Trn_Date = '30-Nov-2013'
32         where Pat_Nbr = 2013;
33
34     commit;
35     select *
36         into treatmentTableRow
37         from Treatment_Table
38         where Pat_Nbr = 2013;
39     DBMS_OUTPUT.PUT_LINE('Read back after change to valid date.');
40     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \ 
at_Nbr);
41     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow \ 
```

```
    .Phys_ID);
39      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \
t_Procedure);
40      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow \
ow.Trt_Date);
41
42      DBMS_OUTPUT.PUT_LINE('Change to invalid date.');
43      update Treatment_Table
44        set Trt_Date = '04-Jan-2013'
45        where Pat_Nbr = 2013;
46      select *
47        into treatmentTableRow
48        from Treatment_Table
49        where Pat_Nbr = 2013;
50      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \
at_Nbr);
51      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow \
.Phys_ID);
52      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \
t_Procedure);
53      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow \
ow.Trt_Date);
54  end;
```

```
1  Error starting at line 1 in command:
2  declare
3      treatmentTableRow Treatment_Table%rowtype;
4  begin
5      treatmentTableRow.Pat_Nbr := 2013;
6      treatmentTableRow.Phys_ID := 101;
7      treatmentTableRow.Tr_Procedure := '13-08';
8      treatmentTableRow.Trt_Date := '04-Nov-2013';
9
10     delete from Treatment_Table
11         where Pat_Nbr = 2013;
12     commit;
13
14     DBMS_OUTPUT.PUT_LINE('Insert with valid date.');
15     insert into Treatment_Table
16         Values treatmentTableRow;
17     commit;
18     select *
19         into treatmentTableRow
20         from Treatment_Table
21         where Pat_Nbr = 2013;
22     DBMS_OUTPUT.PUT_LINE('Read back after insert with valid date.');
23     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \ 
at_Nbr);
24     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow \ 
.Phys_ID);
25     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \ 
t_Procedure);
26     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow \ 
.Trt_Date);
27
28     DBMS_OUTPUT.PUT_LINE('Change to valid date.');
29     update Treatment_Table
30         set Trt_Date = '30-Nov-2013'
31         where Pat_Nbr = 2013;
32     commit;
33     select *
34         into treatmentTableRow
35         from Treatment_Table
36         where Pat_Nbr = 2013;
37     DBMS_OUTPUT.PUT_LINE('Read back after change to valid date.');
38     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \ 
at_Nbr);
```

```
39      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow. \
40      .Phys_ID);
40      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \
t_Procedure);
41      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow. \
ow.Trt_Date);
42
43      DBMS_OUTPUT.PUT_LINE('Change to invalid date.');
44      update Treatment_Table
45          set Trt_Date = '04-Jan-2013'
46          where Pat_Nbr = 2013;
47      select *
48          into treatmentTableRow
49          from Treatment_Table
50          where Pat_Nbr = 2013;
51      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Patient No: ' || treatmentTableRow.P \
at_Nbr);
52      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Physician ID: ' || treatmentTableRow. \
Phys_ID);
53      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatmentTableRow.Tr \
t_Procedure);
54      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Treatment date: ' || treatmentTableRow. \
ow.Trt_Date);
55  end;
56  Error report:
57  ORA-20100: Invalid treatment date.
58  ORA-06512: at "WAQAR1345.CHECK_TREATMENT_DATE", line 4
59  ORA-04088: error during execution of trigger 'WAQAR1345.CHECK_TREATMEN \
T_DATE'
60  ORA-06512: at line 43
61
```

```
1 Insert with valid date.  
2 Trigger gives no error.  
3 Read back after insert with valid date.  
4 Patient No: 2013  
5 Physician ID: 101  
6 Procedure: 13-08  
7 Treatment date: 04-NOV-13  
8 Change to valid date.  
9 Trigger gives no error.  
10 Read back after change to valid date.  
11 Patient No: 2013  
12 Physician ID: 101  
13 Procedure: 13-08  
14 Treatment date: 30-NOV-13  
15 Change to invalid date.  
16 Trigger will generate error.  
17
```

```
1  drop table Trt_Stats;
2  create table Trt_Stats
3    (Trt_Procedure varchar2(5) CONSTRAINT pk_trt_stats PRIMARY KEY,
4     Trt_INS_Count number(3),
5     Trt_DEL_Count number(3),
6     Trt_UPD_Count number(3));
7  Desc Trt_Stats;
```

```
1   table TRT_STATS dropped.
2   table TRT_STATS created.
3 Desc Trt_Stats
4 Name          Null      Type
5 -----
6 TRT_PROCEDURE NOT NULL VARCHAR2(5)
7 TRT_INS_COUNT           NUMBER(3)
8 TRT_DEL_COUNT           NUMBER(3)
9 TRT_UPD_COUNT           NUMBER(3)
10
```

```
1   select *
2   from trt_stats;
```

```
1    no rows selected
2
3
```

```
1  create or replace trigger update_Trt_Stats
2    after insert or update or delete
3    of Trt_Procedure on Treatment_Table
4    for each row
5  declare
6    tv_currentCount_num  number;
7    tv_treatProcUpdated_txt  Treatment_Table.Trt_Procedure%type;
8  begin
9    if inserting then
10      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: conditional pred \
11        icate -> inserting.');
12      select Count(Trt_Procedure)
13        into tv_currentCount_num
14        from Trt_Stats
15        where Trt_Procedure = :NEW.Trt_Procedure;
16      if tv_currentCount_num = 0 then
17        DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: creating new r \
18        ow in Trt_Stats.');
19        insert into Trt_Stats
20          values(:NEW.Trt_Procedure, 1, 0, 0);
21      else
22        DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: Incrementing I \
23        nsert Count in Trt_Stats.');
24        select Trt_INS_Count
25          into tv_currentCount_num
26          from Trt_Stats
27          where Trt_Procedure = :NEW.Trt_Procedure;
28        tv_currentCount_num := tv_currentCount_num + 1;
29        update Trt_Stats
30          set Trt_INS_Count = tv_currentCount_num
31          where Trt_Procedure = :NEW.Trt_Procedure;
32      end if;
33      -- commit;
34    end if;
35
36    if deleting then
37      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: conditional pred \
38        icate -> deleting.');
39      select Count(Trt_Procedure)
40        into tv_currentCount_num
41        from Trt_Stats
42        where Trt_Procedure = :OLD.Trt_Procedure;
43      if tv_currentCount_num = 0 then
```

```
40      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: creating new r \n' \
ow in Trt_Stats.');
41      insert into Trt_Stats
42          values(:OLD.Trт_Procedure, 0, 1, 0);
43  else
44      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: incrementing d \n' \
elete count in Trt_Stats.');
45      select Trт_DEL_Count
46          into tv_currentCount_num
47          from Trt_Stats
48          where Trт_Procedure = :OLD.Trт_Procedure;
49 --      DBMS_OUTPUT.PUT_LINE('Proc: ' || :OLD.Trт_Procedure || ' coun \n' \
t fetched: ' || tv_currentCount_num);
50      tv_currentCount_num := tv_currentCount_num + 1;
51 --      DBMS_OUTPUT.PUT_LINE('Count after increment: ' || tv_currentC \n' \
ount_num);
52      update Trt_Stats
53          set Trт_DEL_Count = tv_currentCount_num
54          where Trт_Procedure = :OLD.Trт_Procedure;
55  end if;
56 --      commit;
57 end if;
58
59 if updating then
60     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: conditional pred \n' \
icate -> updating.');
61     select Count(Trт_Procedure)
62         into tv_currentCount_num
63         from Trt_Stats
64         where Trт_Procedure = :NEW.Trт_Procedure;
65     if tv_currentCount_num = 0 then
66         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: creating new r \n' \
ow in Trt_Stats for new Trт_Procedure value.');
67         insert into Trt_Stats
68             values(:NEW.Trт_Procedure, 0, 0, 1);
69     else
70         DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: incrementing u \n' \
pdate count in Trt_Stats for new Trт_Procedure value.');
71         select Trт_UPD_Count
72             into tv_currentCount_num
73             from Trt_Stats
74             where Trт_Procedure = :NEW.Trт_Procedure;
75         tv_currentCount_num := tv_currentCount_num + 1;
```

```
76      update Trt_Stats
77          set Trt_UPD_Count = tv_currentCount_num
78          where Trt_Procedure = :NEW.Trt_Procedure;
79      end if;
80
81      if :OLD.Trt_Procedure <> :NEW.Trt_Procedure then
82          DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: Trt_Procedure  \
column changed in update.');
83          select Count(Trt_Procedure)
84              into tv_currentCount_num
85              from Trt_Stats
86              where Trt_Procedure = :OLD.Trt_Procedure;
87          if tv_currentCount_num = 0 then
88              DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: creating new  \
row in Trt_Stats for old value.');
89          insert into Trt_Stats
90              values(:OLD.Trt_Procedure, 0, 0, 1);
91          else
92              DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Trigger Message: incremeting  \
update count in Trt_Stats for old value.');
93              select Trt_UPD_Count
94                  into tv_currentCount_num
95                  from Trt_Stats
96                  where Trt_Procedure = :OLD.Trt_Procedure;
97              tv_currentCount_num := tv_currentCount_num + 1;
98              update Trt_Stats
99                  set Trt_UPD_Count = tv_currentCount_num
100                 where Trt_Procedure = :OLD.Trt_Procedure;
101            end if;
102        end if;
103    -- commit;
104    end if;
105
106 end;
```

```
1    TRIGGER UPDATE_TRT_STATS compiled
```

```
1  declare
2      treatmentTableRow Treatment_Table%rowtype;
3      treatStatsTableRow Trt_Stats%rowtype;
4      testTreatProcedure Treatment_Table.Trt_Procedure%type := '12-22';
5  begin
6      treatmentTableRow.Pat_Nbr := 2014;
7      treatmentTableRow.Phys_ID := 101;
8      treatmentTableRow.Trt_Procedure := '13-08';
9      treatmentTableRow.Trt_Date := '04-Nov-2013';
10
11     DBMS_OUTPUT.PUT_LINE('Insert new row into Treatment Table. Treat Pr \
ocedure is ' || treatmentTableRow.Trt_Procedure);
12     insert into Treatment_Table
13         Values treatmentTableRow;
14     commit;
15
16     select *
17         into treatStatsTableRow
18         from Trt_Stats
19         where Trt_Procedure = treatmentTableRow.Trt_Procedure;
20
21     DBMS_OUTPUT.PUT_LINE('Read back from Trt_Stats after insert.');
22     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
23     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
24     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
25     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
26
27     DBMS_OUTPUT.PUT_LINE('Update procedure date. Trt_Procedure column u \
nchanged, no change expected.');
28     update Treatment_Table
29         set Trt_Date = '30-Nov-2013'
30         where Pat_Nbr = treatmentTableRow.Pat_Nbr; --using Pat_Nbr because it will return only 1 row
31     commit;
32
33     select *
34         into treatStatsTableRow
35         from Trt_Stats
36         where Trt_Procedure = treatmentTableRow.Trt_Procedure;
```

```
37
38     DBMS_OUTPUT.PUT_LINE('Read back from Trt_Stats after procedure date \
update. No change in counts.');
39     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
40     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
41     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
42     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
43
44     DBMS_OUTPUT.PUT_LINE('Update Treatment_Table.Trt_Procedure from ' || \
| treatmentTableRow.Trt_Procedure || ' to ' || testTreatProcedure);
45     update Treatment_Table
46         set Trt_Procedure = testTreatProcedure
47         where Pat_Nbr = treatmentTableRow.Pat_Nbr;
48
49     select *
50         into treatStatsTableRow
51         from Trt_Stats
52         where Trt_Procedure = treatmentTableRow.Trt_Procedure;
53     DBMS_OUTPUT.PUT_LINE('Read back from Trt_Stats after Trt_Procedure \
update for old Procedure value.');
54     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
55     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
56     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
57     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
58
59     select *
60         into treatStatsTableRow
61         from Trt_Stats
62         where Trt_Procedure = testTreatProcedure;
63     DBMS_OUTPUT.PUT_LINE('Read back from Trt_Stats after Trt_Procedure \
update for new Procedure value.');
64     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
65     DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
```

```
66      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
67      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
68
69      DBMS_OUTPUT.PUT_LINE('Delete row from Treatment_Table with Procedur \
e No.' || testTreatProcedure);
70      delete from Treatment_Table
71      where Trt_Procedure = testTreatProcedure; --using Pat_Nbr because \
it will return only 1 row
72      commit;
73
74      select *
75      into treatStatsTableRow
76      from Trt_Stats
77      where Trt_Procedure = treatmentTableRow.Trt_Procedure;
78
79      DBMS_OUTPUT.PUT_LINE('Read back Trt_Stats after Treatment_Table row \
deleted, old procedure value.');
80      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
81      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
82      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
83      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
84
85      select *
86      into treatStatsTableRow
87      from Trt_Stats
88      where Trt_Procedure = testTreatProcedure;
89
90      DBMS_OUTPUT.PUT_LINE('Read back Trt_Stats after Treatment_Table row \
deleted, new procedure value.');
91      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
92      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
93      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
94      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
```

```
95
96     DBMS_OUTPUT.PUT_LINE('Insert 5 new rows into Treatment Table. Treat \
Procedure is ' || treatmentTableRow.Trt_Procedure);
97     for i in 1..5 loop
98         treatmentTableRow.Pat_Nbr := treatmentTableRow.Pat_Nbr + 1;
99         insert into Treatment_Table
100            Values treatmentTableRow;
101        end loop;
102        commit;
103
104    DBMS_OUTPUT.PUT_LINE('Delete 3 rows from Treatment Table. Treat Pro \
cedure is ' || treatmentTableRow.Trt_Procedure);
105    for i in 1..3 loop
106        delete from Treatment_Table
107            where Pat_Nbr = treatmentTableRow.Pat_Nbr - i;
108        end loop;
109        commit;
110
111    select *
112        into treatStatsTableRow
113        from Trt_Stats
114        where Trt_Procedure = treatmentTableRow.Trt_Procedure;
115
116    DBMS_OUTPUT.PUT_LINE('Read back from Trt_Stats after insert and del \
etes, old procedure value.');
117    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
118    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableRow. \
ow.Trt_INS_Count);
119    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableRow. \
ow.Trt_DEL_Count);
120    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableRow. \
ow.Trt_UPD_Count);
121
122    select *
123        into treatStatsTableRow
124        from Trt_Stats
125        where Trt_Procedure = testTreatProcedure;
126
127    DBMS_OUTPUT.PUT_LINE('Read back Trt_Stats after inserts and deletes \
, new procedure value.');
128    DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Procedure: ' || treatStatsTableRow. \
Trt_Procedure);
```

```
129      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Insert Count: ' || treatStatsTableR \
ow.Trt_INS_Count);
130      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Delete Count: ' || treatStatsTableR \
ow.Trt_DEL_Count);
131      DBMS_OUTPUT.PUT_LINE(CHR(9) || 'Update Count: ' || treatStatsTableR \
ow.Trt_UPD_Count);
132
133 end;
```

1 anonymous block completed

```
1 Insert new row into Treatment Table. Treat Procedure is 13-08
2   Trigger Message: conditional predicate -> inserting.
3   Trigger Message: creating new row in Trt_Stats.
4 Read back from Trt_Stats after insert.
5   Procedure: 13-08
6   Insert Count: 1
7   Delete Count: 0
8   Update Count: 0
9 Update procedure date. Trt_Procedure column unchanged, no change expec \
ted.
10 Read back from Trt_Stats after procedure date update. No change in cou \
nts.
11   Procedure: 13-08
12   Insert Count: 1
13   Delete Count: 0
14   Update Count: 0
15 Update Treatment_Table.Trt_Procedure from 13-08 to 12-22
16   Trigger Message: conditional predicate -> updating.
17   Trigger Message: creating new row in Trt_Stats for new Trt_Procedure \
value.
18   Trigger Message: Trt_Procedure column changed in update.
19   Trigger Message: incremeting update count in Trt_Stats for old value \
.
20 Read back from Trt_Stats after Trt_Procedure update for old Procedure \
value.
21   Procedure: 13-08
22   Insert Count: 1
23   Delete Count: 0
24   Update Count: 1
25 Read back from Trt_Stats after Trt_Procedure update for new Procedure \
value.
26   Procedure: 12-22
27   Insert Count: 0
28   Delete Count: 0
29   Update Count: 1
30 Delete row from Treatment_Table with Procedure No.12-22
31   Trigger Message: conditional predicate -> deleting.
32   Trigger Message: incrementing delete count in Trt_Stats.
33 Read back Trt_Stats after Treatment_Table row deleted, old procedure v \
alue.
34   Procedure: 13-08
35   Insert Count: 1
36   Delete Count: 0
```

```
37      Update Count: 1
38  Read back Trt_Stats after Treatment_Table row deleted, new procedure v \
alue.
39      Procedure: 12-22
40      Insert Count: 0
41      Delete Count: 1
42      Update Count: 1
43  Insert 5 new rows into Treatment Table. Treat Procedure is 13-08
44      Trigger Message: conditional predicate -> inserting.
45      Trigger Message: Incrementing Insert Count in Trt_Stats.
46      Trigger Message: conditional predicate -> inserting.
47      Trigger Message: Incrementing Insert Count in Trt_Stats.
48      Trigger Message: conditional predicate -> inserting.
49      Trigger Message: Incrementing Insert Count in Trt_Stats.
50      Trigger Message: conditional predicate -> inserting.
51      Trigger Message: Incrementing Insert Count in Trt_Stats.
52      Trigger Message: conditional predicate -> inserting.
53      Trigger Message: Incrementing Insert Count in Trt_Stats.
54  Delete 3 rows from Treatment Table. Treat Procedure is 13-08
55      Trigger Message: conditional predicate -> deleting.
56      Trigger Message: incrementing delete count in Trt_Stats.
57      Trigger Message: conditional predicate -> deleting.
58      Trigger Message: incrementing delete count in Trt_Stats.
59      Trigger Message: conditional predicate -> deleting.
60      Trigger Message: incrementing delete count in Trt_Stats.
61  Read back from Trt_Stats after insert and deletes, old procedure value \
.
62      Procedure: 13-08
63      Insert Count: 6
64      Delete Count: 3
65      Update Count: 1
66  Read back Trt_Stats after inserts and deletes, new procedure value.
67      Procedure: 12-22
68      Insert Count: 0
69      Delete Count: 1
70      Update Count: 1
71
```

```
1  select * from treatment_table;  
2  select * from trt_stats;
```

```
1      PAT_NBR      PHYS_ID TRT_PROCEDURE TRT_DATE
2      -----  -----
3      3249        101 13-08      12-FEB-99
4      1379        103 27-45      25-MAR-99
5      3249        103 88-20      22-JAN-99
6      5116        104 52-14      03-APR-99
7      4500        101 13-08      04-FEB-99
8      8031        102 52-14      15-MAR-00
9      5116        104 52-14      05-FEB-01
10     5872        105 60-00      13-FEB-00
11     3249        103 88-20      24-JAN-00
12     8659        104 60-00      08-APR-01
13     2013        101 13-08      30-NOV-13
14     2015        101 13-08      04-NOV-13
15     2019        101 13-08      04-NOV-13
16
17     13 rows selected
18
19     TRT_PROCEDURE TRT_INS_COUNT TRT_DEL_COUNT TRT_UPD_COUNT
20     -----  -----
21     13-08          6            3            1
22     12-22          0            1            1
23
```