```
SQL> -- Question 1
SQL> -- Find a maximum between three numbers.
SQL> DECLARE
  2      a NUMBER;
  3      b NUMBER;
  4      c NUMBER;
  5      mx NUMBER;
  6  BEGIN
  7      a := &a;
  8      b := &b;
  9      c := &c;
 10
 11      IF a > b THEN
 12          IF a > c THEN
 13              mx := a;
 14          ELSE
 15              mx := c;
 16          END IF;
 17      ELSE
 18          IF b > c THEN
 19              mx := b;
 20          ELSE
 21              mx := c;
 22          END IF;
 23      END IF;
 24
 25      DBMS_OUTPUT.PUT_LINE('Max: ' || mx);
 26  END;
 27  /
Enter value for a: 20
```

```
old   7:     a := &a;
new   7:     a := 20;
Enter value for b: 10
old   8:     b := &b;
new   8:     b := 10;
Enter value for c: 30
old   9:     c := &c;
new   9:     c := 30;
Max: 30

PL/SQL procedure successfully completed.

SQL> -- Question 2
SQL> -- Check whether a number is divisible by 5 and 11 or not.
SQL> DECLARE
  2      a NUMBER;
  3  BEGIN
  4      a := &a;
  5
  6      IF MOD(a, 5) = 0 AND MOD(a, 11) = 0 THEN
  7          DBMS_OUTPUT.PUT_LINE('Divisible by 5 and 11');
  8      ELSE
  9          DBMS_OUTPUT.PUT_LINE('Not divisible by 5 and 11');
 10      END IF;
 11  END;
 12  /
Enter value for a: 55
old   4:     a := &a;
new   4:     a := 55;
Divisible by 5 and 11
```

```
PL/SQL procedure successfully completed.

SQL> -- Question 2
SQL> -- Check whether a number is divisible by 5 and 11 or not.
SQL> DECLARE
  2      a NUMBER;
  3  BEGIN
  4      a := &a;
  5
  6      IF MOD(a, 5) = 0 AND MOD(a, 11) = 0 THEN
  7          DBMS_OUTPUT.PUT_LINE('Divisible by 5 and 11');
  8      ELSE
  9          DBMS_OUTPUT.PUT_LINE('Not divisible by 5 and 11');
 10      END IF;
 11  END;
 12  /
Enter value for a: 47
old   4:     a := &a;
new   4:     a := 47;
Not divisible by 5 and 11

PL/SQL procedure successfully completed.

SQL> -- Question 3
SQL> -- Find the area of rectangle, triangle, and square.
SQL> DECLARE
  2      length NUMBER;
  3      breadth NUMBER;
  4      base NUMBER;
```

```
 5      height NUMBER;
 6      side NUMBER;
 7      area NUMBER;
 8  BEGIN
 9      DBMS_OUTPUT.PUT_LINE('1. Rectangle');
10      length := &length;
11      breadth := &breadth;
12      area := length * breadth;
13      DBMS_OUTPUT.PUT_LINE('Area: ' || area);
14
15      DBMS_OUTPUT.PUT_LINE('2. Triangle');
16      base := &base;
17      height := &height;
18      area := 0.5 * base * height;
19      DBMS_OUTPUT.PUT_LINE('Area: ' || area);
20
21      DBMS_OUTPUT.PUT_LINE('3. Square');
22      side := &side;
23      area := side * side;
24      DBMS_OUTPUT.PUT_LINE('Area: ' || area);
25  END;
26  /
Enter value for length: 3
old  10:     length := &length;
new  10:     length := 3;
Enter value for breadth: 5
old  11:     breadth := &breadth;
new  11:     breadth := 5;
Enter value for base: 8
old  16:     base := &base;
```

```
new  16:     base := 8;
Enter value for height: 4
old  17:     height := &height;
new  17:     height := 4;
Enter value for side: 7
old  22:     side := &side;
new  22:     side := 7;
1. Rectangle
Area: 15
2. Triangle
Area: 16
3. Square
Area: 49

PL/SQL procedure successfully completed.

SQL> DECLARE
  2      phy NUMBER;
  3      chem NUMBER;
  4      bio NUMBER;
  5      math NUMBER;
  6      comp NUMBER;
  7      perc NUMBER;
  8  BEGIN
  9      phy := &phy;
 10      chem := &chem;
 11      bio := &bio;
 12      math := &math;
 13      comp := &comp;
 14
```

```
 15      perc := (phy + chem + bio + math + comp) / 5;
 16
 17      IF perc >= 90 THEN
 18          DBMS_OUTPUT.PUT_LINE('Grade: A');
 19      ELSIF perc >= 80 THEN
 20          DBMS_OUTPUT.PUT_LINE('Grade: B');
 21      ELSIF perc >= 70 THEN
 22          DBMS_OUTPUT.PUT_LINE('Grade: C');
 23      ELSIF perc >= 60 THEN
 24          DBMS_OUTPUT.PUT_LINE('Grade: D');
 25      ELSIF perc >= 40 THEN
 26          DBMS_OUTPUT.PUT_LINE('Grade: E');
 27      ELSE
 28          DBMS_OUTPUT.PUT_LINE('Grade: F');
 29      END IF;
 30  END;
 31  /
Enter value for phy: 74
old   9:      phy := &phy;
new   9:      phy := 74;
Enter value for chem: 83
old  10:      chem := &chem;
new  10:      chem := 83;
Enter value for bio: 93
old  11:      bio := &bio;
new  11:      bio := 93;
Enter value for math: 89
old  12:      math := &math;
new  12:      math := 89;
Enter value for comp: 99
```

```
old  13:     comp := &comp;
new  13:     comp := 99;
Grade: B

PL/SQL procedure successfully completed.

SQL> -- Question 5
SQL> -- Find sum of 100 natural number using loop.
SQL> DECLARE
  2      i NUMBER;
  3      numsum NUMBER;
  4  BEGIN
  5      i := 1;
  6      numsum := 0;
  7
  8      WHILE i <= 100 LOOP
  9          numsum := numsum + i;
 10          i := i + 1;
 11      END LOOP;
 12
 13      DBMS_OUTPUT.PUT_LINE('Sum: ' || numsum);
 14  END;
 15  /
Sum: 5050

PL/SQL procedure successfully completed.

SQL> -- Question 6
SQL> CREATE TABLE empinfo (id NUMBER, name VARCHAR2(20), age NUMBER, address VARCHAR2(20), salary NUMBER);
```

```
Table created.

SQL>
SQL> INSERT INTO empinfo VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000);

1 row created.

SQL> INSERT INTO empinfo VALUES (2, 'Khilan', 25, 'Delhi', 1500);

1 row created.

SQL> INSERT INTO empinfo VALUES (3, 'kaushik', 23, 'Kota', 2000);

1 row created.

SQL> INSERT INTO empinfo VALUES (4, 'Chaitali', 25, 'Mumbai', 6500);

1 row created.

SQL> INSERT INTO empinfo VALUES (5, 'Hardik', 27, 'Bhopal', 8500);

1 row created.

SQL> INSERT INTO empinfo VALUES (6, 'Komal', 22, 'MP', 4500);

1 row created.

SQL>
SQL> ALTER TABLE empinfo ADD CONSTRAINT empinfo_pk PRIMARY KEY (id);
```

```
Table altered.

SQL> -- Question 6.A
SQL> -- Find the name of person having id =1
SQL> DECLARE
  2      e_id empinfo.id%TYPE := 1;
  3      e_name empinfo.name%TYPE;
  4  BEGIN
  5      SELECT name INTO e_name FROM empinfo WHERE id = e_id;
  6      DBMS_OUTPUT.PUT_LINE('The name of person having id=' || e_id || ' is ' || e_name);
  7  END;
  8  /
The name of person having id=1 is Ramesh

PL/SQL procedure successfully completed.

SQL> -- Question 6.B
SQL> -- Find the name, age, salary lives in kota.
SQL> DECLARE
  2      e_city empinfo.address%TYPE := 'Kota';
  3      e_name empinfo.name%TYPE;
  4      e_age empinfo.age%TYPE;
  5      e_salary empinfo.salary%TYPE;
  6  BEGIN
  7      SELECT name, age, salary INTO e_name, e_age, e_salary
  8      FROM empinfo WHERE address = e_city;
  9      DBMS_OUTPUT.PUT_LINE('The name, age, salary who lives in '
 10      || e_city || ' is ' || e_name || ', ' || e_age || ', ' || e_salary);
 11  END;
 12  /
```

```
The name, age, salary who lives in Kota is kaushik, 23, 2000

PL/SQL procedure successfully completed.

SQL>
```