Loops

Basic Loop

```
DECLARE
   i number(1);
   j number(1);
BEGIN
   << outer_loop >>
   FOR i IN 1..3 LOOP
      << inner_loop >>
      FOR j IN 1..3 LOOP
         dbms_output.put_line('i is: '|| i ||
' and j is: ' || j);
      END loop inner_loop;
   END loop outer_loop;
END;
/
```

```
DECLARE
   x number := 10;
BEGIN
   LOOP
      dbms_output.put_line(x);
      x := x + 10;
      IF x > 50 THEN
         exit;
      END IF;
   END LOOP;
   -- after exit, control resumes here
   dbms_output.put_line('After Exit x is: ' || x);
END;
/
```

```
DECLARE
   x number := 10;
BEGIN
   LOOP
      dbms_output.put_line(x);
      x := x + 10;
      exit WHEN x > 50;
   END LOOP;
   -- after exit, control resumes here
   dbms_output.put_line('After Exit x is: ' || x);
END;
/
```

WHILE LOOP

```
DECLARE
   a number(2) := 10;
BEGIN
   WHILE a < 20 LOOP
      dbms_output.put_line('value of a: ' || a);
      a := a + 1;
   END LOOP;
END;
/
```

FOR LOOP

```
DECLARE
   a number(2);
BEGIN
   FOR a in 10 .. 20 LOOP
      dbms_output.put_line('value of a: ' || a);
   END LOOP;
END;
/
```

Reverse FOR LOOP

```
DECLARE
   a number(2) ;
BEGIN
   FOR a IN REVERSE 10 .. 20 LOOP
      dbms_output.put_line('value of a: ' || a);
   END LOOP;
END;
/
```

Nested Loops

```
DECLARE
   i number(3);
   j number(3);
BEGIN
   i := 2;
   LOOP
      j:= 2;
      LOOP
         exit WHEN ((mod(i, j) = 0) or (j = i));
         j := j +1;
      END LOOP;
   IF (j = i ) THEN
```

```
      dbms_output.put_line(i || ' is prime');
   END IF;
   i := i + 1;
   exit WHEN i = 50;
   END LOOP;
END;
/
```

EXIT Statement

```
DECLARE
   a number(2) := 10;
BEGIN
   -- while loop execution
   WHILE a < 20 LOOP
      dbms_output.put_line ('value of a: ' || a);
      a := a + 1;
      IF a > 15 THEN
         -- terminate the loop using the exit statement
         EXIT;
      END IF;
   END LOOP;
END;
/
```

EXIT WHEN

```
DECLARE
   a number(2) := 10;
BEGIN
   -- while loop execution
   WHILE a < 20 LOOP
      dbms_output.put_line ('value of a: ' || a);
      a := a + 1;
      -- terminate the loop using the exit when statement
   EXIT WHEN a > 15;
   END LOOP;
END;
/
```

CONTINUE

```
DECLARE
   a number(2) := 10;
```

```
BEGIN
   -- while loop execution
   WHILE a < 20 LOOP
      dbms_output.put_line ('value of a: ' || a);
      a := a + 1;
      IF a = 15 THEN
         -- skip the loop using the CONTINUE statement
         a := a + 1;
         CONTINUE;
      END IF;
   END LOOP;
END;
/
```

GOTO

```
DECLARE
   a number(2) := 10;
BEGIN
   <<loopstart>>
   -- while loop execution
   WHILE a < 20 LOOP
   dbms_output.put_line ('value of a: ' || a);
      a := a + 1;
      IF a = 15 THEN
         a := a + 1;
         GOTO loopstart;
      END IF;
   END LOOP;
END;
/
```