Exercise 1: Rewrite the following script using a WHILE loop instead of a simple loop.

<table>
<tr><td></td><td>SOLUTION:</td></tr>
<tr><td>SET SERVEROUTPUT ON
DECLARE
v_counter BINARY_INTEGER := 0;
BEGIN
LOOP
-- increment loop counter by one
v_counter := v_counter + 1;
DBMS_OUTPUT.PUT_LINE ('v_counter = '||v_counter);
-- if EXIT condition yields TRUE exit the loop
IF v_counter = 5 THEN
EXIT;
END IF;
END LOOP;
-- control resumes here
DBMS_OUTPUT.PUT_LINE ('Done...');
END;</td><td>SET SERVEROUTPUT ON
DECLARE
   v_counter BINARY_INTEGER := 0;
BEGIN
   WHILE v_counter < 5 LOOP
      --increment loop counter by one
      v_counter := v_counter + 1;
      DBMS_OUTPUT.PUT_LINE('v_counter = '|| v_counter);
   END LOOP;
   --control resumes here
   DBMS_OUTPUT.PUT_LINE('Done...');
END;</td></tr>
</table>

Exercise 2: Rewrite the following script using a numeric FOR loop instead of a WHILE loop.

<table>
<tr><td></td><td>SOLUTION:</td></tr>
<tr><td>SET SERVEROUTPUT ON
DECLARE
v_counter BINARY_INTEGER := 1;
v_sum NUMBER := 0;
BEGIN
WHILE v_counter <= 10 LOOP
v_sum := v_sum + v_counter;
DBMS_OUTPUT.PUT_LINE ('Current sum is: '||v_sum);
-- increment loop counter by one
v_counter := v_counter + 1;
END LOOP;
-- control resumes here
DBMS_OUTPUT.PUT_LINE('The sum of integers between 1'||'and 10 is: '||v_sum);
END;</td><td>SET SERVEROUTPUT ON
DECLARE
   v_sum NUMBER := 0;
BEGIN
   FOR loop_counter IN 1..10 LOOP
      v_sum := v_sum + loop_counter;
      DBMS_OUTPUT.PUT_LINE('Current sum is = '|| v_sum);
   END LOOP;
   --control resumes here
   DBMS_OUTPUT.PUT_LINE('The sum of integers between 1 '||'and 10 is: '||v_sum);
END;</td></tr>
</table>

Exercise 3: Rewrite the following script using a simple loop instead of a numeric FOR loop.

	SOLUTION:

SET SERVEROUTPUT ON
DECLARE
v_factorial NUMBER := 1;
BEGIN
FOR v_counter IN 1..10 LOOP
v_factorial := v_factorial * v_counter;
END LOOP;
-- control resumes here
DBMS_OUTPUT.PUT_LINE ('Factorial of ten is: '||v_factorial);
END;
	SET SERVEROUTPUT ON
	DECLARE
	  v_factorial NUMBER := 1;
	  v_counter NUMBER := 1;
	BEGIN
	  LOOP
	    v_factorial := v_factorial * v_counter;
	    v_counter := v_counter + 1;
	    EXIT WHEN v_counter = 11;
	  END LOOP;
	  DBMS_OUTPUT.PUT_LINE('Factorial of ten is: '||v_factorial);
	END;

Exercise 4: Rewrite the script to calculate the factorial of even integers only between 1 and 10.The script should use a CONTINUE or CONTINUE WHEN statement.

	SOLUTION:

SET SERVEROUTPUT ON
DECLARE
v_factorial NUMBER := 1;
BEGIN
FOR v_counter IN 1..10 LOOP
v_factorial := v_factorial * v_counter;
END LOOP;
-- control resumes here
DBMS_OUTPUT.PUT_LINE
('Factorial of ten is: '||v_factorial);
END;
	SET SERVEROUTPUT ON
	DECLARE
	  v_factorial NUMBER := 1;
	BEGIN
	  FOR loop_counter IN 1..10 LOOP
	    IF MOD(loop_counter,2) != 0 THEN
	      CONTINUE;
	    END IF;
	    v_factorial := v_factorial * loop_counter;
	  END LOOP;
	  DBMS_OUTPUT.PUT_LINE('Factorial of even numbers less or equal to ten is: '||v_factorial);
	END;

Exercise 5: Rewrite the following script using a simple loop instead of the outer FOR loop, and a WHILE loop for the inner FOR loop. Make sure that the output produced by this script does not differ from the output produced by the original script.

```
SET SERVEROUTPUT ON
DECLARE
v_test NUMBER := 0;
BEGIN
<<outer_loop>>
FOR i IN 1..3 LOOP
DBMS_OUTPUT.PUT_LINE('Outer Loop');
DBMS_OUTPUT.PUT_LINE('i = '||i);
DBMS_OUTPUT.PUT_LINE('v_test = '||v_test);
v_test := v_test + 1;
<<inner_loop>>
FOR j IN 1..2 LOOP
DBMS_OUTPUT.PUT_LINE('Inner Loop');
DBMS_OUTPUT.PUT_LINE('j = '||j);
DBMS_OUTPUT.PUT_LINE('i = '||i);
DBMS_OUTPUT.PUT_LINE('v_test = '||v_test);
END LOOP inner_loop;
END LOOP outer_loop;
END;
```

SOLUTION:

```
SET SERVEROUTPUT ON
DECLARE
  v_test NUMBER := 0;
  i NUMBER := 1;
  j NUMBER := 0;
BEGIN
  <<outer_loop>>
  LOOP
    DBMS_OUTPUT.PUT_LINE('OUTER LOOP');
    DBMS_OUTPUT.PUT_LINE('i = '||i);
    DBMS_OUTPUT.PUT_LINE('v_test = '||v_test);
    v_test := v_test + 1;
    j:= 1;
    <<inner_loop>>
    WHILE j<=2 LOOP
      DBMS_OUTPUT.PUT_LINE('INNER LOOP');
      DBMS_OUTPUT.PUT_LINE('j = '||j);
      DBMS_OUTPUT.PUT_LINE('i = '||i);
      DBMS_OUTPUT.PUT_LINE('v_test = '||v_test);
      j := j +1;
    END LOOP inner_loop;
    i := i + 1;
    EXIT WHEN i = 4;
  END LOOP outer_loop;
END;
```

OUTPUT:

```
OUTER LOOP
i = 1
v_test = 0
INNER LOOP
j = 1
i = 1
v_test = 1
INNER LOOP
j = 2
i = 1
v_test = 1
OUTER LOOP
i = 2
v_test = 1
INNER LOOP
```

```
j = 1
i = 2
v_test = 2
INNER LOOP
j = 2
i = 2
v_test = 2
OUTER LOOP
i = 3
v_test = 2
INNER LOOP
j = 1
i = 3
v_test = 3
INNER LOOP
j = 2
i = 3
v_test = 3
```

Exercise 6: Create the following script: Check to see whether there is a record in the STUDENT table for a given student ID. If there is not, insert a record into the STUDENT table for the given student ID.
SOLUTION:

```
SET SERVEROUTPUT ON
DECLARE
   v_student_ID student.student_id%type := &sv_student_ID;
   v_nr NUMBER := 0;
BEGIN
   SELECT COUNT(*)
   INTO v_nr
   FROM student st
   WHERE st.student_id = v_student_ID;
   IF v_nr = 1 THEN
      DBMS_OUTPUT.PUT_LINE('Student with id '||v_student_ID||' exists!');
   ELSE
      INSERT INTO ZIPCODE(zip, city, state, created_by, created_date, modified_by, modified_date)
      VALUES ('51225', 'Arad', 'RO','Albu','11-February-2020','Kokovics','01-March-2020');
      INSERT INTO student st
         (student_id, first_name, last_name, zip, registration_date, created_by, created_date,
modified_by, modified_date)
      VALUES (v_student_ID, 'Vlad', 'Bac', '51225', '01-April-2020','Albu','01-April-2020','Kokovics','01-
April-2020');
   END IF;
   COMMIT;
END;
```

Exercise 7: Create the following script: For a given instructor ID, check to see whether it is assigned to a valid instructor. Then check to see how many sections this instructor teaches, and display this information on the screen.

SOLUTION:

```
SET SERVEROUTPUT ON
DECLARE
   v_instructor_ID instructor.instructor_id%type := &sv_instructor_ID;
   v_nr NUMBER := 0;
   v_nr_sections NUMBER := 0;
BEGIN
   SELECT COUNT(*)
   INTO v_nr
   FROM instructor it
   WHERE it.instructor_id = v_instructor_ID;
   IF v_nr = 1 THEN
      DBMS_OUTPUT.PUT_LINE('Professor with id '||v_instructor_ID||' exists!');
      SELECT COUNT(*)
      INTO v_nr_sections
      FROM section sct
      WHERE sct.instructor_id = v_instructor_ID;
      DBMS_OUTPUT.PUT_LINE('Professor with id '||v_instructor_ID||' teaches '||v_nr_sections||'
sections');
   ELSE
      DBMS_OUTPUT.PUT_LINE('There is NO professor with id '||v_instructor_ID);
   END IF;
END;
```

OUTPUT:

```
Professor with id 1010 exists!
Professor with id 1010 teaches 2 sections
```

SOLUTION2:

```
SET SERVEROUTPUT ON
DECLARE
   v_instructor_id instructor.instructor_id%type := &sv_instructor_id;
   v_name instructor.last_name%type;
   v_nr_sections NUMBER := 0;
BEGIN
   DBMS_OUTPUT.PUT_LINE('Check if a professor with id '||v_instructor_id||' exists!');
   SELECT last_name
   INTO v_name
   FROM instructor
   WHERE instructor_id = v_instructor_id;
   DBMS_OUTPUT.PUT_LINE('The professor with id '||v_instructor_id||' is '||v_name);
```

```
  SELECT COUNT(*)
    INTO v_nr_sections
    FROM section sct
    WHERE sct.instructor_id = v_instructor_ID;
    DBMS_OUTPUT.PUT_LINE('Professor '||v_name||' teaches '||v_nr_sections||' sections');

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('There is no professor with id '||v_instructor_id);
END;
```

Exercise 8: Create the following script: For a course section provided at runtime, determine the number of students registered. If this number is equal to or greater than 10, raise the user-defined exception e_too_many_students and display an error message. Otherwise, display how many students are in a section. Raise a user-defined exception.Otherwise, display how many students are in a section. Make sure your program can process all sections.

SOLUTION: