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

|  |  |
| --- | --- |
|  | SOLUTION: |
| 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; | 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; |

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

|  |  |
| --- | --- |
|  | SOLUTION: |
| 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; | 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; |

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.

|  |  |
| --- | --- |
|  | SOLUTION: |
| 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; | 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: