Exercise 1:

SOLUTION:

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE(SYSDATE);

v\_day VARCHAR2(15);

v\_time TIMESTAMP := SYSTIMESTAMP;

v\_hour VARCHAR2(10);

BEGIN

v\_day := RTRIM(TO\_CHAR(v\_date, 'DAY'));

v\_hour := TO\_CHAR(v\_time,'HH24:MI');

IF v\_day IN ('SATURDAY','SUNDAY') THEN

DBMS\_OUTPUT.PUT\_LINE('Date and time: '||v\_date||' '||v\_time);

IF v\_time BETWEEN '12:01' AND '24:00' THEN

DBMS\_OUTPUT.PUT\_LINE('AFTERNOON!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('MORNING!');

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE(v\_date||' is a working date even though we are in a crisis');

END IF;

DBMS\_OUTPUT.PUT\_LINE('DONE...');

END;

OUTPUT:

24-MAR-20 is a working date even though we are in a crisis

DONE...

Exercise 2:

SOLUTION:

SET SERVEROUTPUT ON

DECLARE

v\_nr\_of\_sections NUMBER := 0;

v\_instructor\_ID NUMBER := &sv\_instructor\_ID;

BEGIN

SELECT COUNT(\*)

INTO v\_nr\_of\_sections

FROM section st

WHERE st.instructor\_id = v\_instructor\_ID;

IF v\_nr\_of\_sections >= 3 THEN

DBMS\_OUTPUT.PUT\_LINE('The Professor needs a vacation!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Prof with ID: '||v\_instructor\_ID||' teaches '||v\_nr\_of\_sections||' section(s).');

END IF;

END;

OUTPUT:

Prof with ID: 1090 teaches 1 section(s).

Exercise 3:

SOLUTION:

SET SERVEROUTPUT ON

DECLARE

v\_date DATE := TO\_DATE(SYSDATE);

v\_day VARCHAR2(15);

v\_time TIMESTAMP := SYSTIMESTAMP;

v\_hour VARCHAR2(10);

BEGIN

v\_day := RTRIM(TO\_CHAR(v\_date, 'DAY'));

v\_hour := TO\_CHAR(v\_time,'HH24:MI');

CASE

WHEN v\_day IN ('SATURDAY' , 'SUNDAY') THEN

DBMS\_OUTPUT.PUT\_LINE('Date and time: '||v\_date||' '||v\_time);

CASE

WHEN v\_time BETWEEN '12:01' AND '24:00' THEN

DBMS\_OUTPUT.PUT\_LINE('AFTERNOON!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('MORNING!');

END CASE;

ELSE DBMS\_OUTPUT.PUT\_LINE('NOT WEEKEND');

END CASE;

DBMS\_OUTPUT.PUT\_LINE('DONE...');

END;

OUTPUT:

NOT WEEKEND

DONE...

Exercise 4:

SOLUTION:

SET SERVEROUTPUT ON

DECLARE

v\_nr\_of\_sections NUMBER := 0;

v\_instructor\_ID NUMBER := &sv\_instructor\_ID;

BEGIN

SELECT COUNT(\*)

INTO v\_nr\_of\_sections

FROM section st

WHERE st.instructor\_id = v\_instructor\_ID;

CASE

WHEN v\_nr\_of\_sections >= 3 THEN

DBMS\_OUTPUT.PUT\_LINE('The Professor needs a vacation!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Prof with ID: '||v\_instructor\_ID||' teaches '||v\_nr\_of\_sections||' section(s)');

END CASE;

END;

OUTPUT:

Prof with ID: 1090 teaches 1 section(s).

Exercise 5:

SOLUTION:

Block 1:

OUTPUT: v\_num is not greater than 0

In this case (IF-THEN-ELSE) the expression evaluates to FALSE, because of the NULL value of the variable, so the STATEMENTS after ELSE will execute, thus it’s statement will execute (it’s message will be printed).

Block2:

OUTPUT: nothing

In this case (IF-THEN) both EXPRESSIONS evaluate to FALSE, same reason as in blobk 1, the variable is assigned value NULL, so their statements will not be evaluated, and the control will be passed to the next executable STATEMENT after the second END IF.

Exercise 6:

SOLUTION:

In the case with:

COALESCE(g.numeric\_grade, e.final\_grade) grade

The “COALESCE” function will compare the 2 inputs, in order, with NULL, and returns the value of the 1st NON-NULL expression, or NULL if all are evaluated to NULL.

IF the values of “g.numeric\_grade,” not NULL, that value will be inserted in the corresponding GRADE column,

Otherwise IF the values of “e.final\_grade,” not NULL, that value will be inserted in the corresponding GRADE column,

Otherwise NULL will be inserted in the corresponding GRADE column.

(this is for every possible row in the table)

In the case with:

NULLIF(g.numeric\_grade, e.final\_grade) grade

The “NULLIF” will compare the 2 input expressions, returning NULL in the case when they are equal, and returning the value of the 1st expression (i.e. the value of g.numeric\_grade) otherwise.

IF the values are equal, than NULL will be inserted in the corresponding GRADE column, otherwise the value of “g.numeric\_grade” will be inserted in the corresponding spot.

(this is for every possible row in the table)