

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 block 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)