

# January 2023 CSE216: Database Sessional

## Online Assignment on PL/SQL

### Subsection: A1

**Time: 30 minutes**

#### Question 1 Required Tables: PREREQ, COURSE, DEPARTMENT

- Write a PL/SQL function that takes two course IDs as arguments and returns **TRUE** if the first course is a prerequisite of strictly more courses than the second one and **FALSE** otherwise.
- For all courses offered by departments in the **Lambeau** building, find whether they have more prerequisites than any course offered by the departments in the **Bronfman** building and print the information using a PL/SQL anonymous block. Order your output by the course titles as shown below. **You cannot use the INTO or WHILE keywords.**

#### Intended Output

```
Heat Transfer needs more prerequisites than Aquatic Chemistry
Heat Transfer needs more prerequisites than C Programming
Heat Transfer needs more prerequisites than Cat Herding
Heat Transfer needs more prerequisites than Existentialism
Heat Transfer needs more prerequisites than International Finance
Heat Transfer needs more prerequisites than International Trade
Heat Transfer needs more prerequisites than Sanitary Engineering
Heat Transfer needs more prerequisites than Strength of Materials
Operating Systems needs more prerequisites than C Programming
Operating Systems needs more prerequisites than Cat Herding
Operating Systems needs more prerequisites than International Finance
Operating Systems needs more prerequisites than International Trade
Operating Systems needs more prerequisites than Strength of Materials
Rock and Roll needs more prerequisites than C Programming
Rock and Roll needs more prerequisites than Cat Herding
Rock and Roll needs more prerequisites than International Finance
Rock and Roll needs more prerequisites than International Trade
Rock and Roll needs more prerequisites than Strength of Materials
```

#### Question 2 Required Tables: TAKES, COURSE

Write a PL/SQL procedure that takes the Student ID and input and calculates the CGPA per every semester (e.g., Fall 2003). If the CGPA chronologically increases (let staying the same be considered an increase) in every semester, print **CGPA is always increasing**. If the CGPA chronologically decreases in every semester, print **CGPA is always decreasing**. Otherwise print **CGPA is following a zig-zag pattern**. You can assume the input student ID will be correct.

For CGPA calculation, you can use the following functions:

```
CREATE OR REPLACE FUNCTION GRADE_TO_POINT(GRADE IN VARCHAR2) RETURN FLOAT IS
```

```

        INVALID_LETTER_GRADE EXCEPTION;
BEGIN
    -- DBMS_OUTPUT.PUT_LINE(GRADE);
    IF GRADE = 'A+' THEN
        RETURN 4.00;
    ELSIF GRADE = 'A ' THEN
        RETURN 3.75;
    ELSIF GRADE = 'A-' THEN
        RETURN 3.50;
    ELSIF GRADE = 'B+' THEN
        RETURN 3.25;
    ELSIF GRADE = 'B ' THEN
        RETURN 3.00;
    ELSIF GRADE = 'B-' THEN
        RETURN 2.75;
    ELSIF GRADE = 'C+' THEN
        RETURN 2.50;
    ELSIF GRADE = 'C ' THEN
        RETURN 2.25;
    ELSIF GRADE = 'C-' THEN
        RETURN 2.15;
    end if;
    RAISE INVALID_LETTER_GRADE;
end;

```

```

create FUNCTION FIND_CGPA(S_ID IN NUMBER) RETURN FLOAT IS
    CGPA FLOAT;
BEGIN
    SELECT ROUND(SUM(GP) / SUM(CREDITS), 4) INTO CGPA
    FROM
        (SELECT MAX(GRADE_TO_POINT(GRADE) * CREDITS) AS GP, CREDITS
         FROM TAKES INNER JOIN COURSE C2 on TAKES.COURSE_ID = C2.COURSE_ID WHERE ID = S_ID
         GROUP BY ID, C2.COURSE_ID, CREDITS);
    -- DBMS_OUTPUT.PUT_LINE('CGPA OF ID ' || S_ID || ' IS ' || CGPA);
    RETURN CGPA;
end;

```