Discussions

This tutorial discussion questions are based on the table `exams(sid, cid, score)` such that:

- Each `sid` is an *INT* and represents a student ID.
- Each `cid` is an *INT* and represents a course ID.
- Each `score` is an *INT* and represents the final exam score of a student in a course.

1. **(Function)** Write a function `max_min` that returns the courses with the largest and smallest score for a given student, with the following properties:
   - It has an input parameter `stu_id`, which is an *INT*.
   - It has two output parameters `max_cid` and `min_cid`, both of which are *INT*.
   - It examines the records in `exams` whose `sid`s are equal to `stu_id`, and identifies the two records among them with the largest and smallest `score`s, respectively.
     - Ties are broken arbitrarily.
     - For the record with the largest `score`, its `cid` is assigned to `max_cid`.
     - For the record with the smallest `score`, if its `score` is smaller than the largest `score`, then its `cid` is assigned to `min_cid`; otherwise, `min_cid` is set to `NULL`.

   The template for `max_min` is provided below:

```
CREATE OR REPLACE FUNCTION max_min
    (IN stu_id INT, OUT max_cid INT, OUT min_cid INT)
RETURNS RECORD AS $$
DECLARE
  max_scr INT;
  min_scr INT;
BEGIN
  -- Write your code here
END;
$$ LANGUAGE plpgsql;
```

2. **(Function)** Write a function `revised_avg` that returns the "*revised average score*" of a given student, with the following properties:

   - It has an input parameter `stu_id`, which is an *INT*.
   - It has one output parameter `r_avg`, which is *NUMERIC*.
   - It examines the records in `exams` whose `sid`s are equal to `stu_id`.
     - If there exists at least 3 such records, the function returns the average score of these records after the following modification:
       * One record with the highest score is excluded (with ties broken arbitrarily).
       * One record with the lowest score is excluded (with ties broken arbitrarily).
     - If there exists fewer than 3 such records, the function returns `NULL`.

   The template for `revised_avg` is provided below:

```
CREATE OR REPLACE FUNCTION revised_avg
    (IN stu_id INT, OUT r_avg NUMERIC)
RETURNS NUMERIC AS $$
  -- Write your code here
$$ LANGUAGE plpgsql;
```

3. **(Cursor?)** Write a function `list_r_avg` that returns the `sid` of each student in `exams` along with his/her revised average score (from 2). For simplicity, we assume that all `sid`s in `exams` are *non-negative* *INT* (*i.e.*, ≥ 0).

   The template for `list_r_avg` is provided below:

```
CREATE OR REPLACE FUNCTION list_r_avg()
RETURNS TABLE (stu_id INT, r_avg NUMERIC) AS $$
DECLARE
  curs CURSOR FOR ( SELECT sid, score FROM exams ORDER BY sid );
  -- Add your variables here
BEGIN
  -- Write your code here
END;
$$ LANGUAGE plpgsql;
```

   You are advised to practice using cursor. Without cursor, we can solve this as follows:

```
SELECT DISTINCT sid, revised_avg(sid)
FROM   exams;
```

   The solution given will recompute the revised average without the use of `revised_avg`.

Challenge

The answers to the following questions is given without explanation. Please discuss them on Canvas.

1. **(Cursor)** Write a function `list_scnd_highest` that returns the `sid` of each student in `exams` along with his/her 2nd highest score (with ties broken arbitrarily). If the student has fewer than 2 scores, then `list_scnd_highest` returns `NULL` as his/her 2nd highest score. For simplicity, we assume that all `sid`s in `exams` are *non-negative* `INT` (*i.e.*, ≥ 0).