Qn2 and 3 uses functions from previous questions.

Qn1.

CREATE OR REPLACE FUNCTION max\_min( IN stu\_id integer, OUT max\_cid integer, OUT min\_cid integer )

RETURNS RECORD as $$

DECLARE

max\_score integer;

min\_score integer;

BEGIN

    SELECT max(score) INTO max\_score

FROM Exams

WHERE sid = stu\_id;

SELECT min(score) INTO min\_score

FROM Exams

WHERE sid = stu\_id;

SELECT cid INTO max\_cid

FROM Exams

WHERE score = max\_score;

IF min\_score < max\_score

THEN

SELECT cid INTO min\_cid

FROM Exams

WHERE score = min\_score;

    ELSE

min\_cid := NULL;

END IF;

END;

$$ LANGUAGE plpgsql;

Qn2.

CREATE OR REPLACE FUNCTION revised\_avg( IN stu\_id integer, OUT r\_avg float )

RETURNS float as $$

DECLARE

count INTEGER;

max\_cid INTEGER;

min\_cid INTEGER;

BEGIN

SELECT count(\*) INTO count

FROM Exams

WHERE sid=stu\_id;

IF count < 3

THEN r\_avg := NULL;

ELSE

SELECT \* FROM max\_min(stu\_id) INTO max\_cid, min\_cid;

SELECT avg(score) INTO r\_avg

        FROM Exams

        WHERE sid=stu\_id AND cid <> max\_cid AND cid <> min\_cid;

    END IF;

END;

$$ LANGUAGE plpgsql;

Qn3.

CREATE OR REPLACE FUNCTION list\_r\_avg()

RETURNS TABLE ( stu\_id integer, ravg float ) AS $$

DECLARE

curs CURSOR FOR (SELECT sid, score from exams order by sid);

current\_student RECORD;

previous\_sid INTEGER;

BEGIN

open curs;

FETCH curs INTO current\_student;

previous\_sid := current\_student.sid;

LOOP

FETCH curs INTO current\_student;

IF NOT FOUND

THEN RETURN QUERY

WITH constant\_id as(

values(previous\_sid)

)

SELECT \* FROM (

constant\_id

CROSS JOIN (

SELECT \* FROM revised\_avg(previous\_sid)

) AS join\_1

) AS row;

EXIT;

END IF;

IF current\_student.sid > previous\_sid

THEN RETURN QUERY

WITH constant\_id as(

values(previous\_sid)

)

SELECT \* FROM (

constant\_id

CROSS JOIN (

SELECT \* FROM revised\_avg(previous\_sid)

) AS join\_1

) AS row;

previous\_sid = current\_student.sid;

END IF;

END LOOP;

END;

$$ LANGUAGE plpgsql;