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;
```