

1. Присвойте порядковый номер каждой записи.

Решение:

```
SELECT *,
        ROW_NUMBER() OVER () AS rn
FROM student_grades;
```

2. Пронумеруйте записи с оценками в порядке их получения каждым студентом.

Решение :

```
SELECT *,
        ROW_NUMBER() OVER (PARTITION BY student_id ORDER BY
exam_date) AS rn
FROM student_grades;
```

3. Добавьте ещё раз записи в таблицу, указав при этом новые идентификационные номера (ID).

```
INSERT INTO student_grades (id, student_name, student_id, subject,
grade, exam_date)
VALUES
(8, 'Петя', 1, 'русский', 4, '2021-06-15'),
(9, 'Петя', 1, 'физика', 5, '2022-01-20'),
(10, 'Петя', 1, 'история', 4, '2022-05-29'),
(11, 'Маша', 2, 'математика', 4, '2021-01-12'),
(12, 'Маша', 2, 'русский', 3, '2022-06-27'),
(13, 'Маша', 2, 'физика', 5, '2022-01-03'),
(14, 'Маша', 2, 'история', 3, '2022-05-12');
```

4. Напишите запрос для нахождения дублей.

Решение :

```
SELECT id
FROM (
    SELECT id,
           ROW_NUMBER() OVER (PARTITION BY exam_date ORDER BY id)
rn
    FROM student_grades
) q1
```

```
WHERE rn > 1;
```

5. Удалите найденные дубли и проверьте результат. Результат будет считаться корректным, если запрос из пункта 4 ничего не выведет.

Решение:

```
DELETE FROM student_grades
WHERE id IN (
  SELECT id
  FROM (
    SELECT id,
           ROW_NUMBER() OVER (PARTITION BY exam_date ORDER BY
id) rn
    FROM student_grades
  ) q1
  WHERE rn > 1
);
```