

Задание

Создать вспомогательную таблицу **applicant**, куда включить **id** образовательной программы, **id** абитуриента, сумму баллов абитуриентов (столбец **itog**) в отсортированном сначала по **id** образовательной программы, а потом по убыванию суммы баллов виде (использовать [запрос](#) из предыдущего урока).

program_id	enrollee_id	itog
1	3	230
1	2	226
1	1	213
2	6	276
2	3	230
2	2	226
3	6	270
3	4	238
3	5	192
3	1	179
4	6	270
4	3	242
4	5	192
4	1	179

Запрос

```
CREATE TABLE applicant AS
SELECT program.program_id, enrollee.enrollee_id, SUM(result) AS itog
FROM
    program
JOIN program_enrollee USING(program_id)
JOIN enrollee USING(enrollee_id)
JOIN program_subject USING(program_id)
JOIN enrollee_subject
    ON enrollee.enrollee_id = enrollee_subject.enrollee_id
    AND enrollee_subject.subject_id = program_subject.subject_id
GROUP BY program_id, enrollee.enrollee_id
ORDER BY program_id, itog DESC;
```

Из таблицы **applicant**, созданной на предыдущем шаге, удалить записи, если абитуриент на выбранную образовательную программу не набрал минимального балла хотя бы по одному предмету (использовать [запрос](#) из предыдущего урока).

program_id	enrollee_id	itog
1	3	230
1	2	226
1	1	213
2	6	276
2	3	230
2	2	226
3	6	270
3	4	238
3	5	192
4	6	270
4	3	242
4	5	192

```
DELETE FROM applicant
WHERE (program_id, enrollee_id) IN (
  SELECT program_subject.program_id, program_enrollee.enrollee_id
  FROM
    program
  JOIN program_subject USING(program_id)
  JOIN program_enrollee USING(program_id)
  JOIN enrollee_subject ON
    program_enrollee.enrollee_id = enrollee_subject.enrollee_id
  AND program_subject.subject_id = enrollee_subject.subject_id
  WHERE enrollee_subject.result < program_subject.min_result);
```

Повысить итоговые баллы абитуриентов в таблице **applicant** на значения дополнительных баллов (использовать [запрос](#) из предыдущего урока).

program_id	enrollee_id	itog
1	3	235
1	2	226
1	1	219
2	6	276
2	3	235
2	2	226
3	6	270
3	4	239
3	5	200
4	6	270
4	3	247
4	5	200

```
UPDATE applicant
  INNER JOIN (
    SELECT enrollee_id, IFNULL(SUM(bonus), 0) AS bonus
  FROM enrollee
  LEFT JOIN enrollee_achievement USING(enrollee_id)
  LEFT JOIN achievement USING(achievement_id)
  GROUP BY enrollee_id) AS bonus_table USING(enrollee_id)
SET itog=itog+bonus_table.bonus
WHERE applicant.enrollee_id=bonus_table.enrollee_id
```

Поскольку при добавлении дополнительных баллов, абитуриенты по каждой образовательной программе могут следовать не в порядке убывания суммарных баллов, необходимо создать новую таблицу **applicant_order** на основе таблицы **applicant**. При создании таблицы данные нужно отсортировать сначала по **id** образовательной программы, потом по убыванию итогового балла. А таблицу **applicant**, которая была создана как вспомогательная, необходимо удалить.

program_id	enrollee_id	itog
1	3	235
1	2	226
1	1	219
2	6	276
2	3	235
2	2	226
3	6	270
3	4	239
3	5	200
4	6	270
4	3	247
4	5	200

```
CREATE TABLE applicant_order AS
SELECT program_id, enrollee_id, itog
FROM applicant
ORDER BY program_id, itog DESC;
DROP TABLE applicant;
```

Включить в таблицу **applicant_order** новый столбец **str_id** целого типа , расположить его перед первым.

```
ALTER TABLE applicant_order ADD str_id INT FIRST
```

applicant_order	
program_id	
enrollee_id	
itog	

str_id	program_id	enrollee_id	itog
Null	1	3	235
Null	1	2	226
Null	1	1	219
Null	2	6	276
Null	2	3	235
Null	2	2	226
Null	3	6	270
Null	3	4	239
Null	3	5	200
Null	4	6	270
Null	4	3	247
Null	4	5	200

Занести в столбец **str_id** таблицы **applicant_order** нумерацию абитуриентов, которая начинается с 1 для каждой образовательной программы.

str_id	program_id	enrollee_id	itog
1	1	3	235
2	1	2	226
3	1	1	219
1	2	6	276
2	2	3	235
3	2	2	226
1	3	6	270
2	3	4	239
3	3	5	200
1	4	6	270
2	4	3	247
3	4	5	200

```

SET @row_num := 0;
SET @prev_program := 0;

UPDATE applicant_order
JOIN (SELECT program_id, enrollee_id,
    (@row_num := IF(@prev_program = program_id, @row_num + 1, 1)) AS str_id,
    (@prev_program := program_id) AS dummy
FROM applicant_order
ORDER BY program_id, itog DESC) AS numbered
ON applicant_order.program_id = numbered.program_id
AND applicant_order.enrollee_id = numbered.enrollee_id
SET applicant_order.str_id = numbered.str_id;

```


Создать таблицу **student**, в которую включить абитуриентов, которые могут быть рекомендованы к зачислению в соответствии с планом набора. Информацию отсортировать сначала в алфавитном порядке по названию программ, а потом по убыванию итогового балла.

name_program	name_enrollee	itog
Математика и компьютерные науки	Степанова Дарья	276
Мехатроника и робототехника	Степанова Дарья	270
Мехатроника и робототехника	Семенов Иван	247
Мехатроника и робототехника	Попов Илья	200
Прикладная математика и информатика	Семенов Иван	235
Прикладная математика и информатика	Абрамова Катя	226
Прикладная механика	Степанова Дарья	270
Прикладная механика	Яковлева Галина	239

```
CREATE TABLE student AS
SELECT name_program, name_enrollee, itog
FROM program
    JOIN applicant_order USING(program_id)
    JOIN enrollee USING(enrollee_id)
WHERE str_id <= plan
ORDER BY name_program, itog DESC;
```

Придумайте один или несколько запросов корректировки данных для предметной области «Абитуриент».

```
SELECT name_program, name_enrollee, itog
FROM program
    JOIN applicant_order USING(program_id)
    JOIN enrollee USING(enrollee_id)
WHERE str_id > plan
ORDER BY name_program;
```

Фрагмент логической схемы базы данных:

