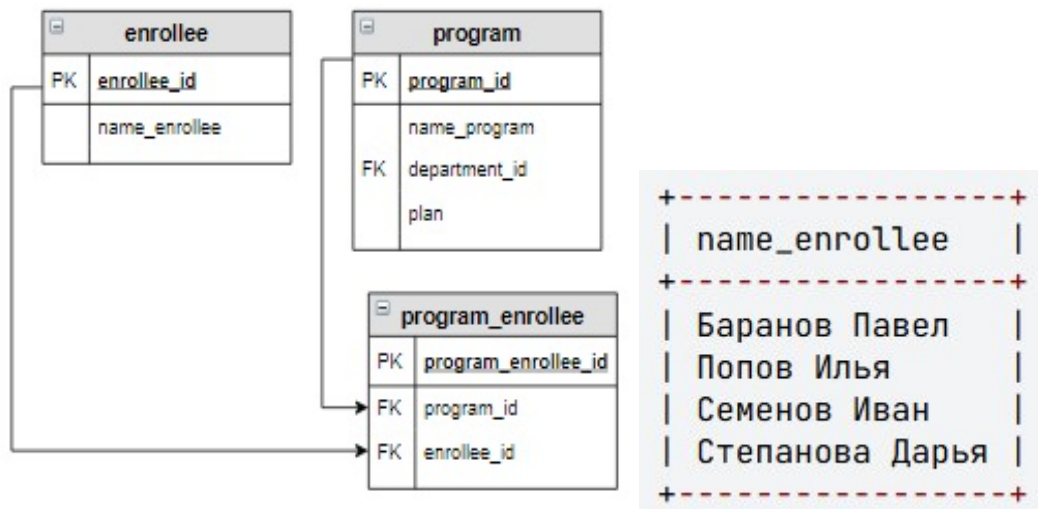


Задание

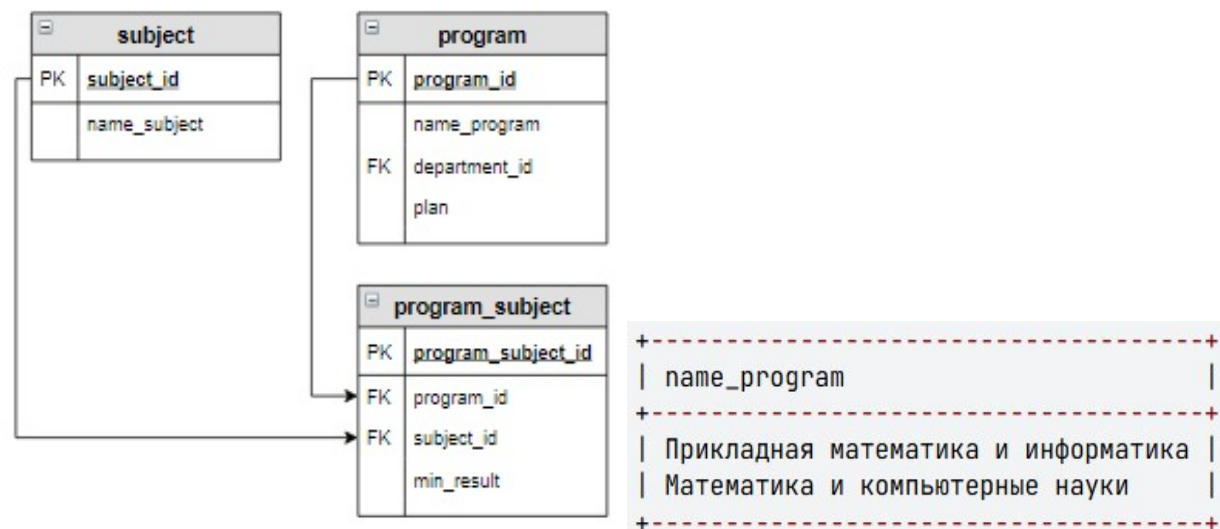
Запрос

Вывести абитуриентов, которые хотят поступать на образовательную программу «Мехатроника и робототехника» в отсортированном по фамилиям виде.



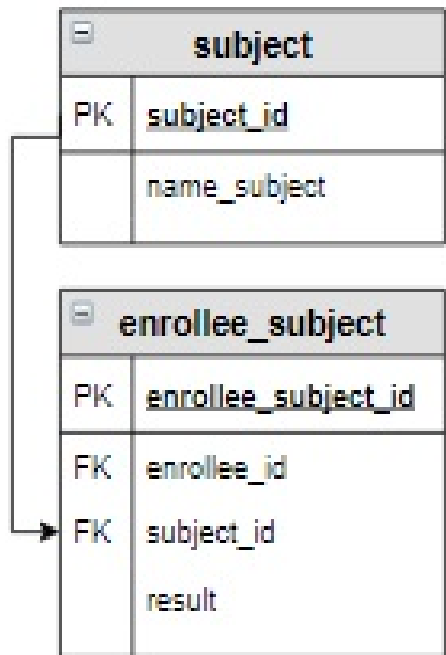
```
SELECT name_enrollee
FROM enrollee
      JOIN program_enrollee USING(enrollee_id)
      JOIN program USING(program_id)
WHERE name_program='Мехатроника и робототехника'
ORDER BY name_enrollee
```

Вывести образовательные программы, на которые для поступления необходим предмет «Информатика». Программы отсортировать в обратном алфавитном порядке.



```
SELECT name_program
FROM program
      JOIN program_subject USING(program_id)
      JOIN subject USING(subject_id)
WHERE name_subject='Информатика'
ORDER BY name_program DESC
```

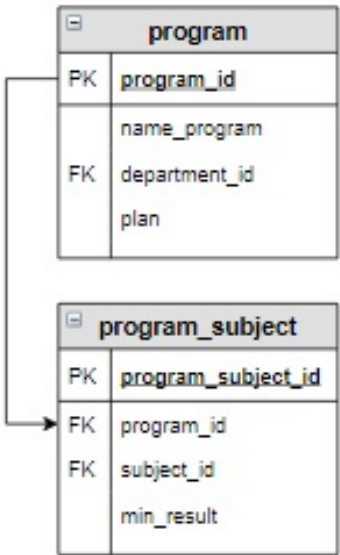
Выведите количество абитуриентов, сдавших ЕГЭ по каждому предмету, максимальное, минимальное и среднее значение баллов по предмету ЕГЭ. Вычисляемые столбцы назвать **Количество**, **Максимум**, **Минимум**, **Среднее**. Информацию отсортировать по названию предмета в алфавитном порядке, среднее значение округлить до одного знака после запятой.



name_subject	Количество	Максимум	Минимум	Среднее
Информатика	4	94	75	82.0
Математика	6	92	67	75.3
Русский язык	6	90	65	77.5
Физика	5	90	41	69.8

```
SELECT name_subject, COUNT(enrollee_id) AS Количество,
MAX(result) AS Максимум, MIN(result) AS Минимум,
ROUND(AVG(result), 1) AS Среднее
FROM subject JOIN enrollee_subject USING(subject_id)
GROUP BY name_subject
ORDER BY name_subject
```

Вывести образовательные программы, для которых минимальный балл ЕГЭ по каждому предмету больше или равен 40 баллам. Программы вывести в отсортированном по алфавиту виде.



name_program
Мехатроника и робототехника
Прикладная математика и информатика

```

SELECT name_program
FROM program JOIN program_subject USING(program_id)
GROUP BY name_program
HAVING MIN(min_result) >=40
ORDER BY name_program
  
```

Вывести образовательные программы, которые имеют самый большой план набора, вместе с этой величиной.

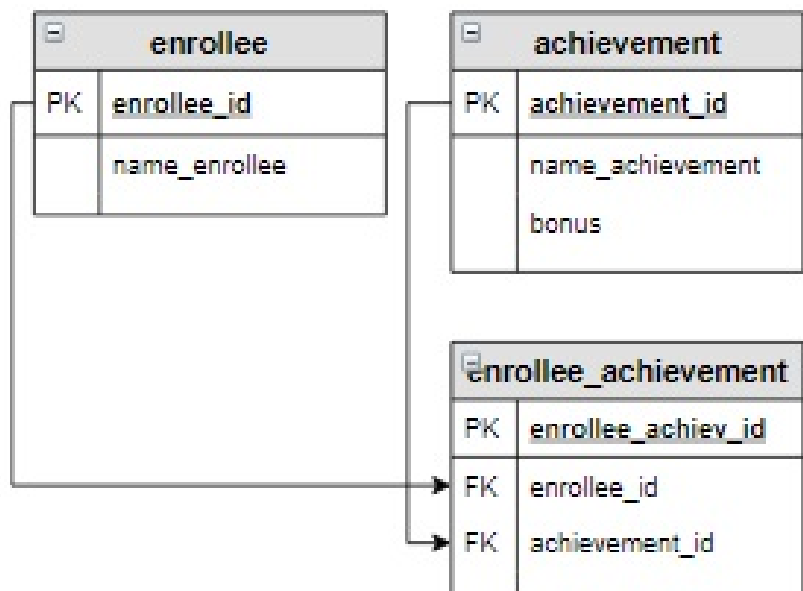
program	
PK	program_id
	name_program
FK	department_id
	plan

name_program	plan
Мехатроника и робототехника	3

```

SELECT name_program, plan
FROM program
WHERE plan=(SELECT MAX(plan) FROM program)
  
```

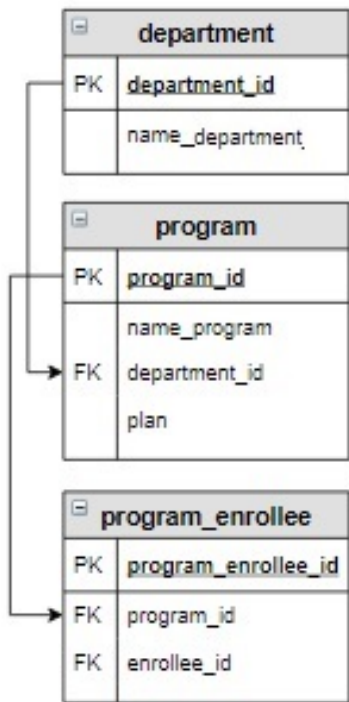
Посчитать, сколько дополнительных баллов получит каждый абитуриент. Столбец с дополнительными баллами назвать **Бонус**. Информацию вывести в отсортированном по фамилиям виде.



name_enrollee	Бонус
Абрамова Катя	0
Баранов Павел	6
Попов Илья	8
Семенов Иван	5
Степанова Дарья	0
Яковлева Галина	1

```
SELECT name_enrollee, IFNULL(SUM(bonus), 0) AS Бонус
FROM
    enrollee
    LEFT JOIN enrollee_achievement USING(enrollee_id)
    LEFT JOIN achievement USING(achievement_id)
GROUP BY name_enrollee
ORDER BY name_enrollee
```


Выведите сколько человек подало заявление на каждую образовательную программу и конкурс на нее (число поданных заявлений деленное на количество мест по плану), округленный до 2-х знаков после запятой. В запросе вывести название факультета, к которому относится образовательная программа, название образовательной программы, план набора абитуриентов на образовательную программу (**plan**), количество поданных заявлений (**Количество**) и **Конкурс**. Информацию отсортировать в порядке убывания конкурса.



name_department	name_program	plan	Количество	Конкурс
Школа естественных наук	Математика и компьютерные науки	1	3	3.00
Инженерная школа	Прикладная механика	2	4	2.00
Школа естественных наук	Прикладная математика и информатика	2	3	1.50
Инженерная школа	Мехатроника и робототехника	3	4	1.33

```
SELECT name_department, name_program, plan,
COUNT(enrollee_id) AS Количество,
ROUND(COUNT(enrollee_id)/plan, 2) AS Конкурс
FROM department
JOIN program USING(department_id)
JOIN program_enrollee USING(program_id)
GROUP BY name_department, name_program, plan
ORDER BY Конкурс DESC
```

Вывести образовательные программы, на которые для поступления необходимы предмет «Информатика» и «Математика» в отсортированном по названию программ виде.

name_program
Математика и компьютерные науки
Прикладная математика и информатика

```
SELECT name_program
FROM
    program
    JOIN program_subject USING(program_id)
    JOIN subject USING(subject_id)
WHERE name_subject='Математика' OR name_subject='Информатика'
GROUP BY name_program
HAVING COUNT(name_subject)=2
ORDER BY name_program
```

Посчитать количество баллов каждого абитуриента на каждую образовательную программу, на которую он подал заявление, по результатам ЕГЭ. В результат включить название образовательной программы, фамилию и имя абитуриента, а также столбец с суммой баллов, который назвать **itog**. Информацию вывести в отсортированном сначала по образовательной программе, а потом по убыванию суммы баллов виде.

name_program	name_enrollee	itog
Математика и компьютерные науки	Степанова Дарья	276
Математика и компьютерные науки	Семенов Иван	230
Математика и компьютерные науки	Абрамова Катя	226
Мехатроника и робототехника	Степанова Дарья	270
Мехатроника и робототехника	Семенов Иван	242
Мехатроника и робототехника	Попов Илья	192
Мехатроника и робототехника	Баранов Павел	179
Прикладная математика и информатика	Семенов Иван	230
Прикладная математика и информатика	Абрамова Катя	226
Прикладная математика и информатика	Баранов Павел	213
Прикладная механика	Степанова Дарья	270
Прикладная механика	Яковлева Галина	238
Прикладная механика	Попов Илья	192
Прикладная механика	Баранов Павел	179

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

Вывести название образовательной программы и фамилию тех абитуриентов, которые подавали документы на эту образовательную программу, но не могут быть зачислены на нее. Эти абитуриенты имеют результат по одному или нескольким предметам ЕГЭ, необходимым для поступления на эту образовательную программу, меньше минимального балла. Информацию вывести в отсортированном сначала по программам, а потом по фамилиям абитуриентов виде.

Например, Баранов Павел по «Физике» набрал 41 балл, а для образовательной программы «Прикладная механика» минимальный балл по этому предмету определен в 45 баллов. Следовательно, абитуриент на данную программу не может поступить.

name_program	name_enrollee
Мехатроника и робототехника	Баранов Павел
Прикладная механика	Баранов Павел

```
SELECT name_program, name_enrollee
FROM
    program
JOIN program_subject USING(program_id)
JOIN program_enrollee USING(program_id)
JOIN enrollee USING(enrollee_id)
JOIN enrollee_subject
ON enrollee.enrollee_id = enrollee_subject.enrollee_id
AND program_subject.subject_id = enrollee_subject.subject_id
WHERE enrollee_subject.result<program_subject.min_result
ORDER BY name_program, name_enrollee
```

Придумайте один или несколько запросов на выборку для предметной области «Абитуриент».

```
SELECT name_enrollee, name_achievement, SUM(result)+bonus AS result
FROM
    enrollee_subject
JOIN enrollee USING(enrollee_id)
JOIN enrollee_achievement USING(enrollee_id)
JOIN achievement USING(achievement_id)
WHERE name_achievement IN ('Золотая медаль', 'Серебряная медаль')
GROUP BY name_enrollee, name_achievement, bonus
ORDER BY name_enrollee
```


Логическая схема базы данных:

