/\*1. Вывести список всех курсов и количество уроков в каждом курсе\*/

select c.name, count(l.id) as numberoflessons

from course c

left join lesson l on c.id = l.course\_id

group by c.name;

/\*2. Вывести список всех учащихся и количество курсов, на которые они записаны\*/

select s.full\_name, count(p.course\_id) as numberofcourses

from student s

left join payment p on s.id = p.student\_id

group by s.full\_name;

/\*3. Вывести список всех учащихся, которые записались на курс "Python для

начинающих"\*/

select s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

where c.name = 'Python для начинающих';

/\*4. Вывести список всех учащихся, которые записались на курс "Веб-разработка с Django"\*/

select s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

where c.name = 'веб-разработка с django';

/\*5. Вывести список учащихся, которые записались на курсы с названием,

содержащим слово "Python", и у которых средний балл по всем урокам выше 4\*/

select s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join grade g on s.id = g.student\_id

where c.name like '%Python%'

group by s.full\_name

having avg(g.grade) > 4;

/\*6. Вывести список курсов, на которые записались все учащиеся, и количество

уроков в каждом курсе\*/

select c.name, count(l.id) as numberoflessons

from course c

left join lesson l on c.id = l.course\_id

group by c.name;

/\*7. Вывести список учащихся, которые записались на курсы, стоимость которых

выше средней стоимости всех курсов, и у которых есть уведомления\*/

with avgcourseprice as (

select avg(price) as avgprice

from course

)

select s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join notification n on s.id = n.student\_id

where c.price > (select avgprice from avgcourseprice);

/\*8. Вывести список курсов, на которые записались учащиеся с email-адресами,

содержащими слово "example", и количество учащихся на каждом курсе\*/

select c.name, count(distinct p.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join student s on p.student\_id = s.id

where s.email like '%example%'

group by c.name;

/\*9. Вывести список учащихся, которые записались на курсы, у которых есть уроки с

названием, содержащим слово "Введение", и у которых есть оценки выше 4\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join grade g on s.id = g.student\_id and l.id = g.lesson\_id

where l.name like '%Введение%' and g.grade > 4;

/\*10. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки ниже 3, и количество таких учащихся на каждом курсе\*/

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

where g.grade < 3

group by c.name;

/\*11. Вывести список учащихся, которые записались на курсы, у которых есть уроки

с материалами, содержащими слово "PDF", и у которых есть уведомления\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join notification n on s.id = n.student\_id

where l.materials like '%PDF%';

/\*12. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки выше 4, и количество таких учащихся на каждом курсе\*/

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

where g.grade > 4

group by c.name;

/\*13. Вывести список учащихся, которые записались на курсы, у которых есть уроки

с названием, содержащим слово "Основы", и у которых есть уведомления\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join notification n on s.id = n.student\_id

where l.name like '%Основы%';

/\*14. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки выше 4, и количество таких учащихся на каждом курсе, отсортированный по

количеству учащихся в порядке убывания\*/

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

where g.grade > 4

group by c.name

order by numberofstudents desc;

/\*15. Вывести список учащихся, которые записались на курсы, у которых есть уроки

с материалами, содержащими слово "видео", и у которых есть уведомления,

отсортированный по дате последнего уведомления\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join notification n on s.id = n.student\_id

where l.materials like '%видео%'

order by n.notification\_date desc;

/\*16. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки выше 4, и количество таких учащихся на каждом курсе, отсортированный по

количеству учащихся в порядке убывания, и вывести только те курсы, на которые

записалось более 2 учащихся\*/

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

where g.grade > 4

group by c.name

having count(distinct g.student\_id) > 2

order by numberofstudents desc;

/\*17. Вывести список учащихся, которые записались на курсы, у которых есть уроки

с названием, содержащим слово "Основы", и у которых есть уведомления,

отсортированный по дате последнего уведомления\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join notification n on s.id = n.student\_id

where l.name like '%Основы%'

order by n.notification\_date desc;

/\*18. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки выше 4, и количество таких учащихся на каждом курсе, отсортированный по

количеству учащихся в порядке убывания, и вывести только те курсы, на которые

записалось более 2 учащихся, и стоимость которых выше средней стоимости всех

курсов\*/

with avgcourseprice as (

select avg(price) as avgprice

from course

)

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

where g.grade > 4 and c.price > (select avgprice from avgcourseprice)

group by c.name

having count(distinct g.student\_id) > 2

order by numberofstudents desc;

/\*19. Вывести список учащихся, которые записались на курсы, у которых есть уроки

с материалами, содержащими слово "PDF", и у которых есть уведомления,

отсортированный по дате последнего уведомления, и вывести только тех учащихся, у

которых есть оценки выше 4\*/

select distinct s.full\_name

from student s

join payment p on s.id = p.student\_id

join course c on p.course\_id = c.id

join lesson l on c.id = l.course\_id

join notification n on s.id = n.student\_id

join grade g on s.id = g.student\_id

where l.materials like '%PDF%' and g.grade > 4

order by n.notification\_date desc;

/\*20. Вывести список курсов, на которые записались учащиеся, у которых есть

оценки выше 4, и количество таких учащихся на каждом курсе, отсортированный по

количеству учащихся в порядке убывания, и вывести только те курсы, на которые

записалось более 2 учащихся, и стоимость которых выше средней стоимости всех

курсов, и у которых есть уроки с материалами, содержащими слово "видео\*/

with avgcourseprice as (

select avg(price) as avgprice

from course

)

select c.name, count(distinct g.student\_id) as numberofstudents

from course c

join payment p on c.id = p.course\_id

join grade g on p.student\_id = g.student\_id

join lesson l on c.id = l.course\_id

where g.grade > 4 and c.price > (select avgprice from avgcourseprice) and l.materials like '%видео%'

group by c.name

having count(distinct g.student\_id) > 2

order by numberofstudents desc;