

| Books | | | | | Orders | | | | |
|-------|--------|---------|-------|-------|--------|-----|--------|-----------|--------|
| id | name | genre | pages | price | | id | name | id_book | date |
| 1 | name_1 | genre_1 | 100 | 10 | | 1 | name_1 | id_book_1 | date_1 |
| 2 | name_2 | genre_2 | 200 | 20 | | 2 | name_2 | id_book_2 | date_2 |
| 3 | name_3 | genre_3 | 300 | 30 | | 3 | name_3 | id_book_3 | date_3 |
| ... | ... | ... | ... | ... | | ... | ... | ... | ... |

В идеале бы знать на каких колонках какой таблицы присутствуют индексы, т.к. это на прямую будет воздействовать на быстродействие отработки запроса
Я же всегда перед выполнением запроса смотрю на explain, что помогает оптимизировать запросы и не получить по шапке от девопсов

Составить SQL код для следующих задач:

1. Вывести самую объемную книгу

```
select
  b.name,
  b.pages
from
  books b
order by
  b.pages desc limit 1
```

2. Вывести всех студентов, что брали книгу name_1 (студент мог 1 книгу брать несколько раз)

```
select distinct
  o.name
from
  orders o
inner join
  books b on b.id = o.id_book and b.name = 'name_1'
```

3. Какое кол-во студентов брали книгу name_2 (поле orders.name - это уникальный ключ идентификатор студента)

без group by и join всей таблицы

```
select
  b.name book_title,
  (
    select
      count(distinct o.name)
    from
      orders o
    where
      o.id_book = b.id
  ) unique_students_cnt
from
  books b
where
  b.name = 'name_2'
```

с группировкой и join

```
select
  b.name book_title,
  count(distinct o.name) unique_students_cnt
from
  books b
inner join
  orders o on o.book_id = b.id and b.name = 'name_2'
group_by
  b.name
```

4. Вывести в алфавитном порядке названия самых дорогих книг в каждом жанре. С условием, что все цены уникальны
не уверен на счет правильности, но логику задания старался выразить запросом.

```
select
  b_outer.name,
  b_outer.genre,
  b_outer.price
from books b_outer
inner join (
select
  b_inner.genre,
  max(b_inner.price) price
from
  books b_inner
group by
  b.genre)
  max_genre_prices mgp on mgp.genre = b_outer.genre and mgp.price = b_outer.price
order by
  b_outer.name ASC
```

5. Вывести данные по книгам, в названии которых присутствует символ "%"

```
select
  b.*
from
  books b
where
  b.name like '\%' escape '\'
```

6. Вывести имена последних трех студентов и книги, которые они брали когда либо

```
with prep as (  
  select  
    orders.name,  
    books.name,  
    orders.date,  
    ROW_NUMBER () OVER (ORDER BY date desc) as rn  
  from  
    orders  
  left join  
    books on books.id=ordersid_book  
)  
select * from prep where rn<3
```

7. Вывести книги, которые студенты не брали в течении текущего месяца

```
select  
  *  
from  
  books b  
where b.id not in (  
  select  
    o.book_id  
  from  
    orders o  
  where  
    month(o.date) = month(current_date())  
    and year(o.date) = year(current_date())  
)
```