

## Задание

## Запрос

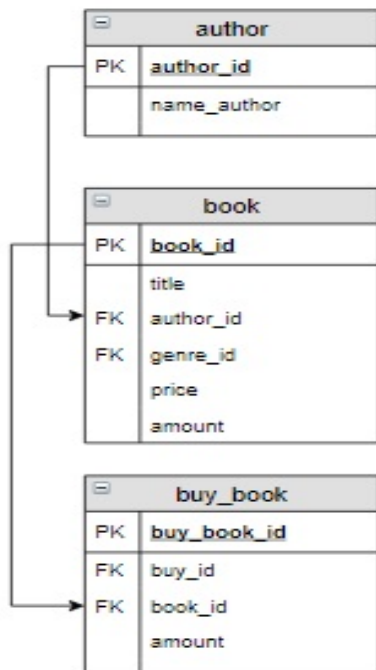
Вывести все заказы Баранова Павла (**id** заказа, какие книги, по какой цене и в каком количестве он заказал) в отсортированном по номеру заказа и названиям книг виде.

buy_id	title	price	amount
1	Идиот	460.00	1
1	Мастер и Маргарита	670.99	1
1	Черный человек	570.20	2
4	Игрок	480.50	1

```
SELECT DISTINCT buy.buy_id, book.title, book.price, buy_book.amount
FROM
    client
    INNER JOIN buy ON client.client_id = buy.client_id
    INNER JOIN buy_book ON buy_book.buy_id = buy.buy_id
    INNER JOIN book ON buy_book.book_id=book.book_id
WHERE client.name_client = 'Баранов Павел'
ORDER BY buy.buy_id, book.title;
```

Посчитать, сколько раз была заказана каждая книга, для книги вывести ее автора (нужно посчитать, в каком количестве заказов фигурирует каждая книга). Вывести фамилию и инициалы автора, название книги, последний столбец назвать **Количество**. Результат отсортировать сначала по фамилиям авторов, а потом по названиям книг.

```
SELECT author.name_author, book.title, COUNT(
    buy_book.buy_book_id) AS Количество
FROM
    author
    LEFT JOIN book ON author.author_id=book.author_id
    LEFT JOIN buy_book ON book.book_id=buy_book.book_id
WHERE book.title IS NOT NULL
GROUP BY author.name_author, book.title
ORDER BY author.name_author, book.title;
```



name_author	title	Количество
Булгаков М.А.	Белая гвардия	1
Булгаков М.А.	Мастер и Маргарита	2
Достоевский Ф.М.	Братья Карамазовы	0
Достоевский Ф.М.	Игрок	1
Достоевский Ф.М.	Идиот	2
Есенин С.А.	Стихотворения и поэмы	0
Есенин С.А.	Черный человек	1
Пастернак Б.Л.	Лирика	1

Вывести города, в которых живут клиенты, оформлявшие заказы в интернет-магазине. Указать количество заказов в каждый город, этот столбец назвать **Количество**. Информацию вывести по убыванию количества заказов, а затем в алфавитном порядке по названию городов.

name_city	Количество
Владивосток	2
Москва	1
Санкт-Петербург	1

```
SELECT city.name_city, COUNT(buy.buy_id) AS Количество
FROM
    city
    INNER JOIN client ON city.city_id=client.city_id
    INNER JOIN buy ON client.client_id=buy.client_id
GROUP BY city.name_city
ORDER BY Количество DESC, city.name_city
```

Вывести номера всех оплаченных заказов и даты, когда они были оплачены.

buy_id	date_step_end
1	2020-02-20
2	2020-02-28
3	2020-03-05

```
SELECT buy.buy_id, buy_step.date_step_end
FROM
    buy
    INNER JOIN buy_step ON buy.buy_id=buy_step.buy_id
    INNER JOIN step ON buy_step.step_id=step.step_id
WHERE step.name_step='Оплата' AND buy_step.date_step_end IS NOT NULL
```

Вывести информацию о каждом заказе: его номер, кто его сформировал (фамилия пользователя) и его стоимость (сумма произведений количества заказанных книг и их цены), в отсортированном по номеру заказа виде. Последний столбец назвать **Стоимость**.

buy_id	name_client	Стоимость
1	Баранов Павел	2271.39
2	Семенов Иван	1037.98
3	Абрамова Катя	2131.49
4	Баранов Павел	480.50

```
SELECT buy.buy_id, name_client, SUM(
    buy_book.amount*price) AS Стоимость
FROM
    client
    INNER JOIN buy ON client.client_id=buy.client_id
    INNER JOIN buy_book ON buy.buy_id=buy_book.buy_id
    INNER JOIN book ON buy_book.book_id=book.book_id
GROUP BY buy_id
```

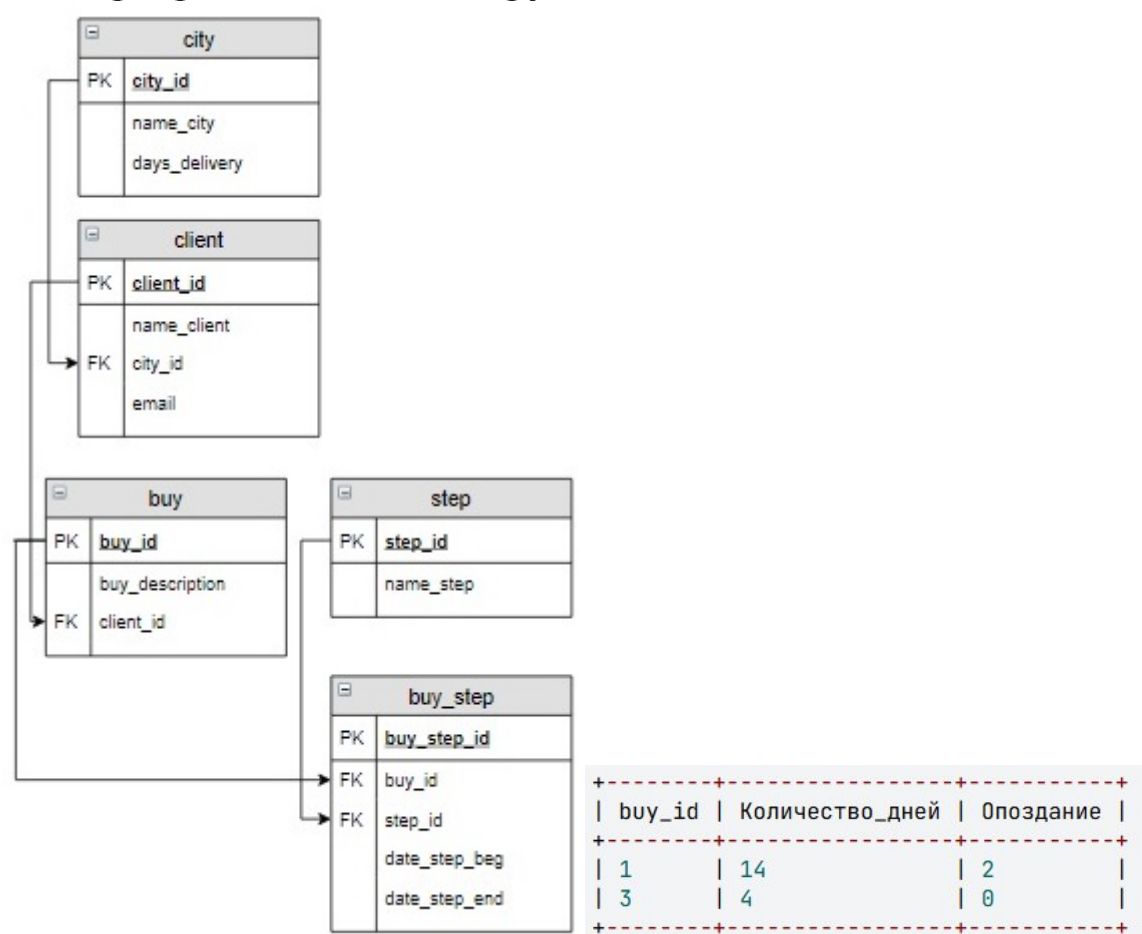
Вывести номера заказов (**buy\_id**) и названия этапов, на которых они в данный момент находятся. Если заказ доставлен – информацию о нем не выводить. Информацию отсортировать по возрастанию **buy\_id**.



buy_id	name_step
2	Транспортировка
3	Доставка
4	Оплата

```
SELECT buy.buy_id, name_step
FROM
    buy
    INNER JOIN buy_step ON buy.buy_id=buy_step.buy_id
    INNER JOIN step ON buy_step.step_id=step.step_id
WHERE date_step_beg IS NOT NULL AND date_step_end IS NULL
ORDER BY buy_id
```

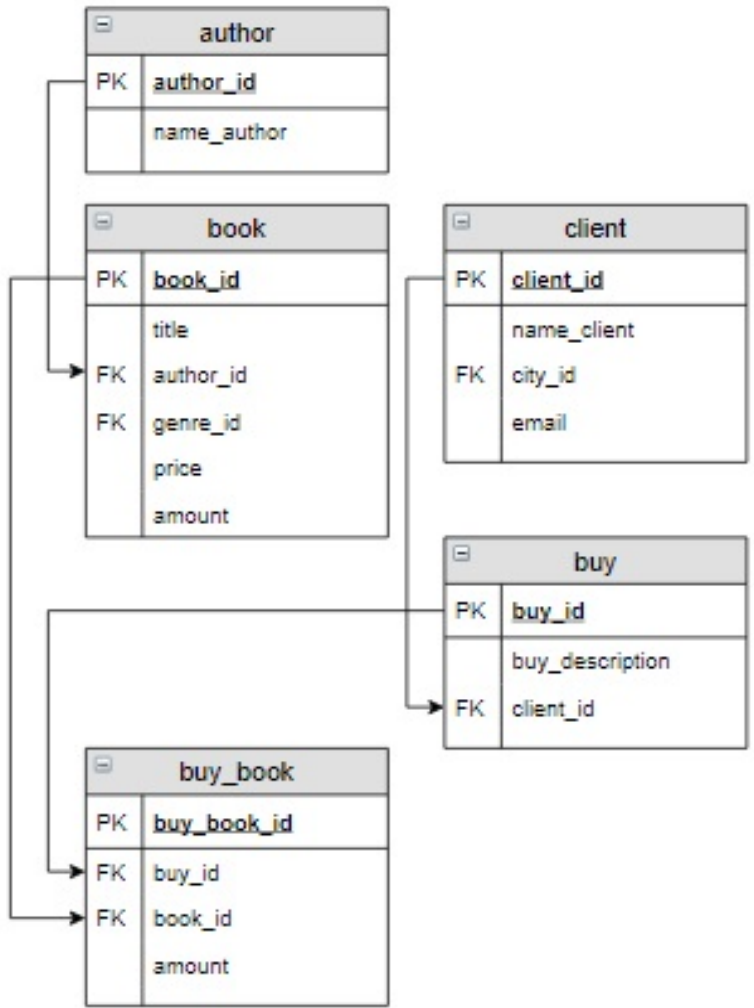
В таблице **city** для каждого города указано количество дней, за которые заказ может быть доставлен в этот город (рассматривается только этап **Транспортировка**). Для тех заказов, которые прошли этап транспортировки, вывести количество дней за которое заказ реально доставлен в город. А также, если заказ доставлен с опозданием, указать количество дней задержки, в противном случае вывести 0. В результат включить номер заказа (**buy\_id**), а также вычисляемые столбцы **Количество\_дней** и **Опоздание**. Информацию вывести в отсортированном по номеру заказа виде.



```
SELECT buy.buy_id, DATEDIFF(
    buy_step.date_step_end, buy_step.date_step_beg) AS Количество_дней,
IF(DATEDIFF(
    buy_step.date_step_end, buy_step.date_step_beg) > city.days_delivery,
    DATEDIFF(
        buy_step.date_step_end, buy_step.date_step_beg) - city.days_delivery, 0) AS Опоздание
FROM
    step
INNER JOIN buy_step ON step.step_id=buy_step.step_id
INNER JOIN buy ON buy_step.buy_id=buy.buy_id
INNER JOIN client ON buy.client_id=client.client_id
INNER JOIN city ON client.city_id=city.city_id
WHERE buy_step.date_step_end IS NOT NULL AND step.name_step='Транспортировка'
ORDER BY buy.buy_id
```



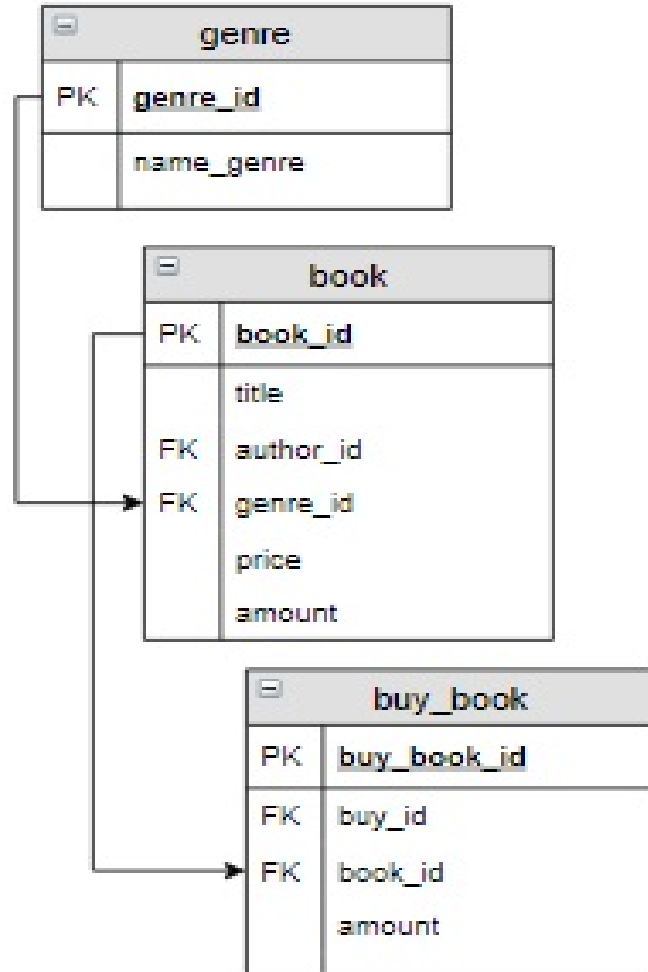
Выбрать всех клиентов, которые заказывали книги Достоевского, информацию вывести в отсортированном по алфавиту виде. В решении используйте фамилию автора, а не его **id**.



name_client
Абрамова Катя
Баранов Павел

```
SELECT DISTINCT name_client
FROM
    author
    INNER JOIN book USING (author_id)
    INNER JOIN buy_book USING (book_id)
    INNER JOIN buy USING (buy_id)
    INNER JOIN client USING (client_id)
WHERE name_author LIKE 'Достоевский%'
ORDER BY name_client
```

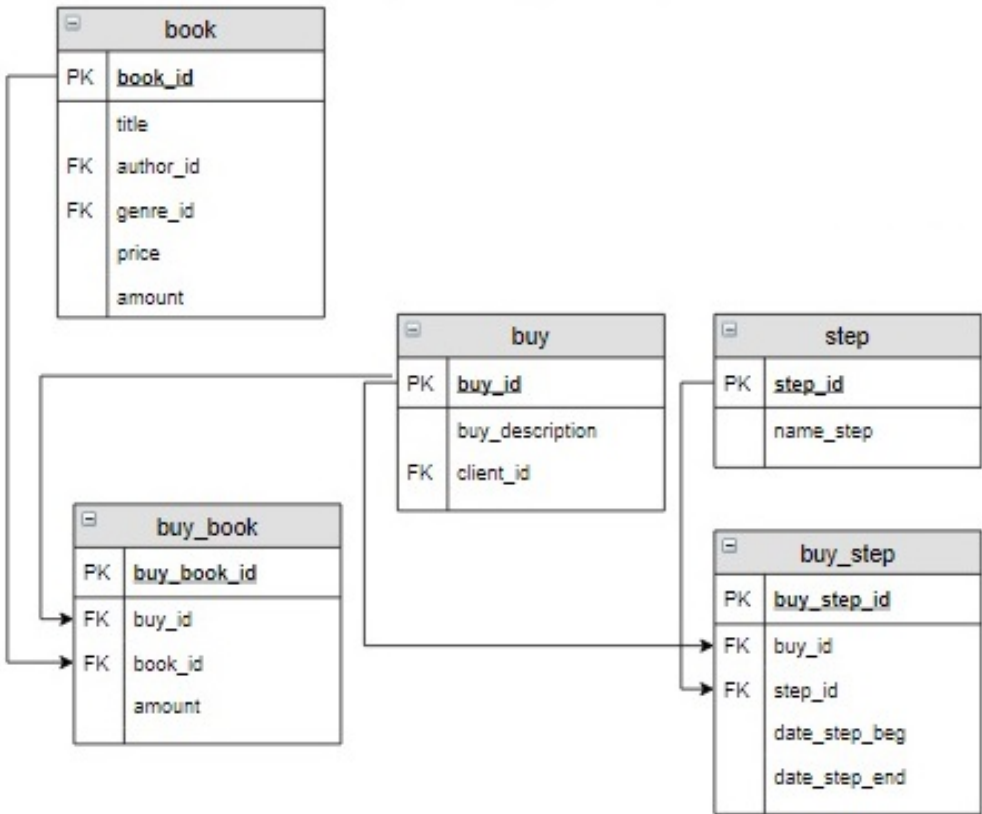
Вывести жанр (или жанры), в котором было заказано больше всего экземпляров книг, указать это количество. Последний столбец назвать **Количество**.



name_genre	Количество
Роман	7

[illegible]

Сравнить ежемесячную выручку от продажи книг за текущий и предыдущий годы. Для этого вывести год, месяц, сумму выручки в отсортированном сначала по возрастанию месяцев, затем по возрастанию лет виде. Название столбцов: **Год, Месяц, Сумма**.



Год	Месяц	Сумма
2019	February	5626.30
2020	February	3309.37
2019	March	6857.50
2020	March	2131.49

```
SELECT YEAR(date_payment) AS Год,
MONTHNAME(date_payment) AS Месяц,
SUM(amount * price) AS Сумма
FROM buy_archive
GROUP BY MONTHNAME(date_payment), YEAR(date_payment)
UNION ALL
SELECT YEAR(date_step_end) AS Год,
MONTHNAME(date_step_end) AS Месяц,
SUM(buy_book.amount * price) AS Сумма
FROM buy_book
    INNER JOIN book USING(book_id)
    INNER JOIN buy_step USING(buy_id)
    INNER JOIN step USING(step_id)
WHERE date_step_end IS NOT NULL AND name_step = 'Оплата'
GROUP BY MONTHNAME(date_step_end), YEAR(date_step_end)
ORDER BY Месяц, Год;
```

Для каждой отдельной книги необходимо вывести информацию о количестве проданных экземпляров и их стоимости за 2020 и 2019 год . За 2020 год проданными считать те экземпляры, которые уже оплачены. Вычисляемые столбцы назвать **Количество** и **Сумма**. Информацию отсортировать по убыванию стоимости.

title	Количество	Сумма
Братья Карамазовы	8	6247.20
Мастер и Маргарита	6	4024.38
Идиот	5	2281.80
Белая гвардия	3	1581.10
Черный человек	2	1140.40
Лирика	2	1037.98
Игрок	2	961.80
Стихотворения и поэмы	1	650.00

```
SELECT title, SUM(amount) AS Количество,
SUM(Сумма) AS Сумма
FROM (SELECT book.title, buy_archive.amount,
buy_archive.amount * buy_archive.price AS Сумма
FROM buy_archive
INNER JOIN book USING(book_id)
UNION ALL
SELECT book.title, buy_book.amount,
buy_book.amount * book.price AS Сумма
FROM buy_book
INNER JOIN book USING(book_id)
INNER JOIN buy_step USING(buy_id)
INNER JOIN step USING(step_id)
WHERE date_step_end IS NOT NULL
AND name_step = 'Оплата') query_in
GROUP BY title
ORDER BY Сумма DESC;
```

Придумайте один или несколько запросов на выборку для предметной области «Интернет-магазин книг» (в таблицы занесены данные, как на [первом шаге](#) урока).

```
SELECT name_client, name_city, buy_id
FROM
step
INNER JOIN buy_step USING (step_id)
INNER JOIN buy USING (buy_id)
INNER JOIN client USING (client_id)
INNER JOIN city USING (city_id)
WHERE date_step_end IS NULL AND name_step='Доставка'
AND date_step_beg IS NOT NULL
ORDER BY buy_id
```



# Логическая схема базы данных:

