

**Задание 1. Выведите для каждого покупателя его адрес, город и страну проживания.**

**Ожидаемый результат запроса:** <https://ibb.co/hLdvkfx>

**Ответ**

```
select concat(c.first_name, ' ', c.last_name) "Фамилия и имя", a.address  
"Адрес", ci.city "Город", c2.country "Страна"  
from customer c  
left join address a on a.address_id = c.address_id  
left join city ci on ci.city_id = a.city_id  
left join country c2 on c2.country_id = ci.country_id
```

**+ Скриншот результата**

**Задание 2. С помощью SQL-запроса посчитайте для каждого магазина количество его покупателей.**

**Ожидаемый результат запроса:** <https://ibb.co/CKWT86p>

**Доработайте запрос и выведите только те магазины, у которых количество покупателей больше 300. Для решения используйте фильтрацию по сгруппированным строкам с функцией агрегации.**

**Ожидаемый результат запроса:** <https://ibb.co/5Y976Mc>

**Доработайте запрос, добавив в него информацию о городе магазина, фамилии и имени продавца, который работает в нём.**

**Ожидаемый результат запроса:** <https://ibb.co/phg71Z8>

**Ответ**

```
select store_id "ID магазина", count(customer_id) "Количество  
покупателей"  
from customer c  
group by store_id
```

Скриншот 2\_1

```
select store_id "ID магазина", count(customer_id) "Количество  
покупателей"  
from customer c  
group by store_id  
having count(customer_id) > 300
```

Скриншот 2\_2

```
select c.store_id "ID магазина", count(c.customer_id) "Количество  
покупателей", c2.city "Город магазина", concat(s.last_name, ' ',  
s.first_name) "Фамилия и имя продавца"  
from customer c , staff s  
left join address a on a.address_id = s.address_id  
left join city c2 on c2.city_id = a.city_id  
where c.store_id = s.store_id  
group by c.store_id, c2.city, concat(s.last_name, ' ', s.first_name)  
having count(c.customer_id) > 300
```

Скриншот 2\_3

### Ошибка

**В задании 2.3 нужно вывести город нахождения магазина, а не город проживания сотрудника, внимательней к джойнам.**

### Исправление 2.3

```
select  
    c.store_id "ID магазина",  
    count(c.customer_id) "Количество покупателей",  
    c2.city "Город магазина",  
    concat(s.last_name, ' ', s.first_name) "Фамилия и имя продавца"  
from customer c , staff s  
left join store s2 on s2.store_id = s.store_id  
left join address a on s2.address_id = a.address_id  
left join city c2 on c2.city_id = a.city_id
```

```
where c.store_id = s.store_id  
group by 1, 3, 4  
having count(c.customer_id) > 300
```

Скрин 2\_3\_1

Комментарий: насколько я понял, в прошлый раз я присоединял  
сотрудник -> адрес -> город

Теперь я пошел путем магазин -> адрес -> город

**Задание 3. Выведите топ-5 покупателей, которые взяли в аренду  
за всё время наибольшее количество фильмов.**

Ожидаемый результат запроса: <https://ibb.co/H21FPns>

**Ответ:**

```
select concat(c.last_name, ' ', c.first_name) "Фамилия и имя покупателя",  
count(rental_id) "Количество фильмов"  
from rental r, customer c  
where c.customer_id = r.customer_id  
group by 1  
order by count(rental_id)  
desc limit 5
```

Скриншот 3\_

**Задание 4. Посчитайте для каждого покупателя 4 аналитических  
показателя:**

- количество взятых в аренду фильмов;
- общую стоимость платежей за аренду всех фильмов (значение округлите до целого числа);
- минимальное значение платежа за аренду фильма;
- максимальное значение платежа за аренду фильма.

Ожидаемый результат запроса: <https://ibb.co/kKw33Pc>

Ответ

```
select concat(c.last_name, ' ', c.first_name) "Фамилия и имя покупателя",  
       count(rental_id) "Количество фильмов",  
       sum(round(amount)) "Общая стоимость платежей",  
       min(amount) "Минимальная стоимость платежей",  
       max(amount) "Максимальная стоимость платежей"  
from payment p  
left join customer c on c.customer_id = p.customer_id  
group by 1
```

Скриншот 4\_

### Ошибка

**В 4 задании в запросе отсутствуют данные по аренде, таким образом гарантировать корректность данных по количеству фильмов нельзя. Добавьте в запрос отношение rental.**

### Добавил:

```
select concat(c.last_name, ' ', c.first_name) "Фамилия и имя покупателя",  
       count(rental_id) "Количество фильмов",  
       sum(round(amount)) "Общая стоимость платежей",  
       min(amount) "Минимальная стоимость платежей",  
       max(amount) "Максимальная стоимость платежей"  
from payment p  
left join rental r on r.rental_id = p.rental_id  
left join customer c on c.customer_id = r.customer_id  
group by 1
```

Теперь цепочка payment -> rental -> customer

Скрин 4\_1

**Задание 5.** Используя данные из таблицы городов, составьте одним запросом всевозможные пары городов так, чтобы в результате не было пар с одинаковыми названиями городов. Для решения необходимо использовать декартово произведение. Ожидаемый результат запроса: <https://ibb.co/X71Lhk5>

**Ответ**

```
select c.city "Город 1", c2.city "Город 2"
from city c
cross join city c2
where c.city != c2.city
```

Скриншот 5\_

**Задание 6.** Используя данные из таблицы rental о дате выдачи фильма в аренду (поле rental\_date) и дате возврата (поле return\_date), вычислите для каждого покупателя среднее количество дней, за которые он возвращает фильмы. Ожидаемый результат запроса: <https://ibb.co/KX5PKKh>

**Ответ**

```
select r.customer_id, round(avg(extract(epoch from r.return_date -
r.rental_date)/86400)::numeric,2)
from rental r
group by r.customer_id
order by 1
```

Скриншот 6\_

**Ошибка**

**В 6 задании можно просто привести к типу date и не работать с секундами.**

**Исправил**

```
select
  r.customer_id "ID покупателя",
  round(avg(r.return_date :: date - r.rental_date :: date), 2) as "Среднее
количество дней на возврат"
from rental r
group by 1
order by 1
```

Скрин 6\_1