



№1

**Условие:** Получи все данные из таблицы city.

**Решение:** SELECT \* FROM city

№2

**Условие:** Получи имя (first\_name), отчество (middle\_name) и фамилию (last\_name) из таблицы сотрудников (employee).

**Решение:** SELECT first\_name, middle\_name, last\_name FROM employee

№3

**Условие:** Напиши запрос, возвращающий одну строку с тремя столбцами со значениями 1, 2 и 3. Назначь столбцам псевдонимы соответственно one, TWO и "Three". Используй двойные кавычки только для "Three".

**Решение:** SELECT 1 AS one, 2 AS TWO, 3 AS "Three"

№4

**Условие:** Напиши запрос для получения из таблицы city 3-х столбцов данных:

1. Текущее время на сервере. Назови столбец server\_date.
2. name - название города. Назови столбец name.
3. Название города в верхнем регистре (необходимо воспользоваться функцией upper). Назови столбец name\_upper.

В последующих заданиях требования будут оформлены в более лаконичном виде, например текущее задание в таком виде будет выглядеть так:

1. server\_date - текущее время на сервере;
2. name - название города;
3. name\_upper - Название города в верхнем регистре (необходимо воспользоваться функцией upper).

**Решение:** SELECT now() as server\_date, name, upper(name) as name\_upper FROM city

№5

**Условие:** Из таблицы employee получи один столбец full\_name в формате 'first\_name last\_name'

**Решение:** SELECT first\_name || ' ' || last\_name as full\_name FROM employee

№6

**Условие:** Из таблицы цены на товар в магазинах (product\_price) получи следующую информацию

1. product\_id - идентификатор продукта;
2. price - цена;
3. price\_sale - половина цены товара

**Решение:** SELECT product\_id,  
price,  
price/2 AS price\_sale  
FROM product\_price

№7

**Условие:** Из таблицы товаров заказа purchase\_item получи список уникальных идентификаторов продуктов, которые когда-либо заказывали.

**Решение:** SELECT DISTINCT product\_id FROM purchase\_item

№8

**Условие:** Получи все столбцы таблицы категории товаров category, у которых идентификатор родительской категории равен 2.

**Решение:** SELECT \*  
FROM category  
WHERE parent\_category\_id = 2

№9

**Условие:**Получи информацию о магазинах из таблицы store\_address, находящихся в городе с идентификатором 6 по адресу 'ул. Чкалова, 42'  
Выведи все поля из таблицы.

**Решение:**SELECT \*  
FROM store\_address  
WHERE city\_id = 6  
AND address = 'ул. Чкалова, 42'

№10

**Условие:**Получи все поля из таблицы сотрудников, у которых имя Виктор или отчество Иванович.

**Решение:** SELECT \*  
  
FROM employee  
WHERE first\_name = 'Виктор'  
OR middle\_name = 'Иванович'

№11

**Условие:** Получи все поля из таблицы сотрудников, кроме тех, у кого имя Виктор или отчество Иванович.

Исключи сотрудников с незаполненным отчеством

**Решение:**SELECT \* FROM employee  
  
WHERE NOT (first\_name = 'Виктор'  
OR middle\_name = 'Иванович')

№12

**Условие:** Получи все поля из таблицы сотрудников employee с именем Сергей, у которых идентификатор должности один из 'CEO', 'DIRECTOR', 'GENERAL\_MANAGER'.

**Решение:**SELECT \*  
FROM employee  
WHERE (rank\_id = 'CEO'

```
OR rank_id = 'DIRECTOR'  
OR rank_id = 'GENERAL_MANAGER')  
AND first_name = 'Сергей'
```

#### №13

**Условие:** Получи информацию об адресах магазина с кодом 100, находящихся не в Москве (city\_id = 1).

Выведи все поля из таблицы store\_address.

**Решение:** SELECT \*

```
FROM store_address  
WHERE store_id = 100  
AND city_id != 1
```

#### №14

**Условие:** Напиши запрос для получения информации о продуктах со стоимостью, попадающей в один из диапазонов: от 5000 до 15000 и от 30000 до 40000 включая концы отрезков.

Выведи все поля из таблицы product\_price.

**Решение:** SELECT \*

```
FROM product_price  
WHERE price BETWEEN 5000 AND 15000  
OR price BETWEEN 30000 AND 40000
```

#### №15

**Условие:** Напиши запрос для получения информации о сотрудниках employee занимающих должности, отличные от 'SELLER', 'MANAGER'.

Выведи все поля из таблицы employee.

**Решение:** SELECT \*

```
FROM employee  
WHERE rank_id NOT IN ('SELLER', 'MANAGER')
```

#### №16

**Условие:** Напиши запрос для получения информации о сотрудниках, фамилия которых начинается на 'М' и заканчивается на 'В':

1. first\_name - имя сотрудника;
2. last\_name - фамилия сотрудника.

**Решение:** SELECT first\_name, last\_name

```
FROM employee  
WHERE last_name LIKE 'М%В'
```

#### №17

**Условие:** Получи информацию об адресах магазинов, в которых заполнены и часы работы, и телефон.

Выведи все поля из таблицы store\_address.

**Решение:** SELECT \*  
FROM store\_address  
WHERE opening\_hours IS NOT NULL  
AND phone IS NOT NULL

№18

**Условие:** Получи всю информацию о городах (city), отсортированную по названию.

**Решение:** SELECT \*  
FROM city  
ORDER BY name

№19

**Условие:** Получи информацию о сотрудниках, отсортированную сначала по идентификатору магазина, затем по должности в алфавитном порядке, а затем по фамилии в порядке, противоположном алфавитному:

1. employee\_id - идентификатор сотрудника;
2. store\_id - идентификатор магазина;
3. rank\_id - идентификатор должности;
4. last\_name - фамилия сотрудника.

**Решение:** SELECT employee\_id,  
store\_id,  
rank\_id,  
last\_name  
FROM employee  
ORDER BY store\_id, rank\_id, last\_name DESC

№20

**Условие:** Получи информацию о стоимости первых 5 товаров с максимальной ценой. Выведи все поля из таблицы product\_price.

**Решение:** SELECT \*  
FROM product\_price  
ORDER BY price DESC  
LIMIT 5

№21

**Условие:** Получи информацию о стоимости первых 5 товаров с максимальной ценой, пропустив первые 2. Выведи все поля из таблицы product\_price.

**Решение:** SELECT \*  
FROM product\_price  
ORDER BY price DESC  
LIMIT 5  
OFFSET 2

#### №22

**Условие:** К таблице товаров product присоедини таблицу категорий товаров category по идентификатору категории товаров. Обрати внимание на полученные столбцы в результате

**Решение:** SELECT \*  
FROM product  
JOIN category  
ON category.category\_id = product.category\_id

#### №23

**Условие:** К таблице заказов purchase присоедини таблицу товаров заказа purchase\_item по идентификатору заказа purchase\_id. Выведи следующие данные:

1. purchase\_id - идентификатор заказа;
2. purchase\_date - дата заказа;
3. store\_id - идентификатор магазина;
4. product\_id - идентификатор товара;
5. total\_price - итоговая стоимость, равная количеству единиц товара умноженную на цену за единицу товара.

Отсортируй результат сначала по идентификатору магазина, затем по дате заказа по убыванию.

**Решение:** SELECT p.purchase\_id, p.purchase\_date, p.store\_id, pi.product\_id, pi.count \*  
pi.price as total\_price  
FROM purchase p  
JOIN purchase\_item pi  
ON pi.purchase\_id = p.purchase\_id  
ORDER BY p.store\_id, p.purchase\_date DESC

#### №24

**Условие:** Получи информацию о товарах заказов, которых купили одновременно 5 или более штук:

1. purchase\_id - идентификатор заказа;
2. purchase\_date - дата заказа;
3. product\_id - идентификатор товара;
4. count - количество единиц товара заказа;
5. price - цена за единицу товара.

Отсортируй результат сначала по количеству единиц товара по убыванию, затем по дате заказа по убыванию.

**Решение:** SELECT p.purchase\_id,  
p.purchase\_date,  
pi.product\_id,  
pi.count,  
pi.price  
FROM purchase p  
JOIN purchase\_item pi  
ON pi.purchase\_id = p.purchase\_id

WHERE pi.count >= 5  
ORDER BY pi.count DESC, p.purchase\_date DESC

#### №25

**Условие:** Помогите сотрудникам найти повышение в должности, либо перейти на ту же должность, но в другой магазин.

Для каждого сотрудника получите информацию о должностях либо магазина, в котором работает сотрудник, либо должностях с таким же идентификатором должности в другом магазине. Выведите следующие столбцы:

- last\_name - фамилия сотрудника;
- first\_name - имя сотрудника;
- store\_id\_employee - идентификатор магазина сотрудника;
- store\_id\_rank - идентификатор магазина должности;
- rank\_id - идентификатор должности в магазине;
- rank\_name - название должности.

Отсортируйте результат по

1. фамилии;
2. имени;
3. идентификатору магазина должности;
4. идентификатору должности.

**Решение:** SELECT e.last\_name, e.first\_name,  
  
e.store\_id AS store\_id\_employee,  
r.store\_id AS store\_id\_rank,  
r.rank\_id,  
r.name AS rank\_name  
FROM employee e  
JOIN rank r  
ON r.store\_id = e.store\_id  
OR (r.rank\_id = e.rank\_id AND r.store\_id != e.store\_id)  
ORDER BY e.last\_name, e.first\_name, r.store\_id, r.rank\_id

#### №26

**Условие:** Для каждой должности rank получите информацию о других должностях этого же магазина. Выведите:

1. store\_id - идентификатор магазина;
2. rank\_id - идентификатор должности;
3. rank\_id\_other - идентификатор другой должности магазина.

Отсортируйте результат по store\_id, rank\_id, rank\_id\_other.

**Решение:** SELECT r.store\_id, r.rank\_id,  
ro.rank\_id AS rank\_id\_other  
FROM rank r  
JOIN rank ro

```
ON ro.store_id = r.store_id
AND ro.rank_id != r.rank_id
ORDER BY r.store_id, r.rank_id, ro.rank_id
```

#### №27

**Условие:** Получи информацию о товарах категории товаров:

- category\_name - название категории товаров;
- product\_name - название товара.

Если в категории нет товаров, то значения столбца product\_name должно быть равно null.

Отсортируй результат сначала по названию категории товаров, затем по названию товара.

**Решение:** SELECT c.name AS category\_name,  
p.name AS product\_name  
FROM category c  
LEFT JOIN  
product p  
ON p.category\_id = c.category\_id  
ORDER BY c.name, p.name

#### №28

**Условие:** Получи информацию о городах часовых поясов:

- time\_offset - время относительно UTC;
- city\_name - название города.

Если в часовом поясе нет городов, то city\_name должно принимать значение null. Используй для этого RIGHT JOIN.

Отсортируй результат по:

1. time\_offset;
2. city\_name.

**Решение:** SELECT t.time\_offset, c.name AS city\_name  
FROM city c  
RIGHT JOIN  
timezone t  
ON c.timezone\_id = t.timezone\_id  
ORDER BY t.time\_offset, city\_name

#### №29

**Условие:** Сравни таблицы product\_price и purchase\_item и выведи уникальные идентификаторы товаров, которые больше не продаются, но их покупали раньше (отсутствует запись в product\_price, но присутствует в purchase\_item), либо товар ни разу не продавался (отсутствует запись в purchase\_item, но присутствует в product\_price). Выведи столбцы:



- product\_id\_price - идентификатор товара из product\_price;
- product\_id\_purchase - идентификатор товара из purchase\_item.

**Решение:** SELECT DISTINCT  
 pp.product\_id AS product\_id\_price,  
 pi.product\_id AS product\_id\_purchase  
 FROM product\_price pp  
 FULL JOIN  
     purchase\_item pi  
 ON pp.product\_id = pi.product\_id  
 WHERE pp.product\_id IS NULL  
 OR pi.product\_id IS NULL

#### №30

**Условие:** Получи декартово произведение строк таблиц часовых поясов и городов и выведи столбцы:

- city\_name - название города;
- timezone\_id\_city - поле timezone\_id из таблицы city;
- timezone\_id\_timezone - поле timezone\_id из таблицы timezone;
- time\_offset - время относительно UTC.

Отсортируй результат по названию города, затем по коду временной зоны из таблицы timezone.

**Решение:**  
 SELECT c.name AS city\_name,  
       c.timezone\_id AS timezone\_id\_city,  
       t.timezone\_id AS timezone\_id\_timezone,  
       t.time\_offset  
 FROM city c,  
       timezone t  
 ORDER BY c.name, t.timezone\_id

#### №31

**Условие:** Получи информацию о товарах категории товаров с идентификатором 6 не используя конструкцию JOIN:

1. category\_name - название категории товаров;
2. product\_name - название товара;
3. description - описание товара.

Отсортируй результат по названию товара.

**Решение:** SELECT c.name AS category\_name,  
       p.name as product\_name,  
       p.description  
 FROM category c,  
       product p  
 WHERE p.category\_id = c.category\_id  
 AND c.category\_id = 6

ORDER BY p.name

№32

**Условие:** Получи следующую информацию по таблице цен на товары product\_price:

- price\_min - минимальная цена товара;
- price\_avg - средняя цена товара;
- price\_max - максимальная цена товара.

**Решение:** SELECT min(price) AS price\_min,  
avg(price) AS price\_avg,  
max(price) AS price\_max  
FROM product\_price

№33

**Условие:** Из таблицы заказов получи единственную строку с информацией:

1. count\_total - общее количество заказов;
2. count\_employee - количество заказов, которые оформили сотрудники магазина.\

**Решение:** SELECT count (\*) AS count\_total,  
  
count (employee\_id) AS count\_employee  
FROM purchase

№34

**Условие:** Из таблицы заказов получи информацию:

1. count\_total - общее количество заказов;
2. count\_employee - количество заказов, которые оформили сотрудники магазина;
3. count\_distinct\_employee - количество сотрудников магазинов, когда-либо оформивших заказ.

**Решение:** SELECT count(\*) AS count\_total,  
count(employee\_id) AS count\_employee,  
count(DISTINCT employee\_id) AS count\_distinct\_employee  
FROM purchase