

SQL СОБЕСЕДОВАНИЕ

18 задач с ответами и проверками

```
WITH paid AS (  
    SELECT client_id, SUM(total) revenue  
    FROM orders WHERE status = 'paid'  
    GROUP BY client_id  
)  
SELECT * FROM paid  
ORDER BY revenue DESC;
```

JOIN · GROUP BY · CTE · окна

NULL · даты · подзапросы · проверки

Глеб Зайцев

**SQL для собеседования.
18 задач с ответами**

«Автор»

2026

Зайцев Г.

SQL для собеседования. 18 задач с ответами / Г. Зайцев —
«Автор», 2026

Практическая книга для подготовки к SQL-собеседованию без обещаний трудоустройства и чудесных результатов. Внутри 18 задач на маленьких таблицах: фильтры, JOIN, GROUP BY, HAVING, COUNT DISTINCT, NULL, даты, CTE, подзапросы, UNION, оконные функции, ранжирование и итоговый отчёт. Все примеры запускаются в SQLite, данные встроены прямо в книгу.

© Зайцев Г., 2026

© Автор, 2026

Содержание

Перед первой задачей	5
Как устроены хорошие SQL-задачи на интервью	5
Как тренироваться	6
Маршрут по задачам	7
Практические задачи	8
Задача 1. Посчитать оплаченные заказы	8
Задача 2. Найти второй по величине чек	9
Задача 3. Найти клиентов без заказов	10
Конец ознакомительного фрагмента.	11

Глеб Зайцев

SQL для собеседования.

18 задач с ответами

Перед первой задачей

Как устроены хорошие SQL-задачи на интервью

На собеседовании редко проверяют знание одной функции в вакууме. Обычно смотрят, понимаете ли вы зерно данных, фильтры, связи таблиц и смысл метрики.

Эта книга не обещает оффер. Она даёт короткую практику на типовых мыслительных ходах: посчитать, соединить, найти пропуск, убрать дубль и объяснить результат.

Каждая задача маленькая, но проверяемая. Сначала попробуйте предсказать ответ глазами, затем сравните его с SQL-запросом и объяснением.

Как тренироваться

Не переписывайте решение механически. Перед запуском назовите словами, какая строка является фактом и почему именно эти фильтры стоят в запросе.

Если задача кажется простой, усложните её контрольным вопросом: что изменится при дублирующем заказе, NULL, отмене или отсутствующей связи.

Примеры запускаются в SQLite. Для других СУБД отдельные функции могут называться иначе, но логика отчёта остаётся той же.

Маршрут по задачам

- [1. Посчитать оплаченные заказы](#)
- [2. Найти второй по величине чек](#)
- [3. Найти клиентов без заказов](#)
- [4. Найти дубли email](#)
- [5. Посчитать месячную выручку](#)
- [6. Найти категорию с максимальной выручкой](#)
- [7. Посчитать выручку клиента через CTE](#)
- [8. Учесть NULL-скидку](#)
- [9. Найти товары без продаж через NOT EXISTS](#)
- [10. Разбить чеки на сегменты](#)
- [11. Пронумеровать заказы клиента](#)
- [12. Посчитать накопительную выручку](#)
- [13. Найти клиентов выше среднего](#)
- [14. Сохранить нулевые категории после JOIN](#)
- [15. Посчитать уникальных покупателей](#)
- [16. Объединить лиды из двух каналов](#)
- [17. Выбрать заказы в полуоткрытом периоде](#)
- [18. Собрать итоговый отчёт кандидата](#)

Практические задачи

Задача 1. Посчитать оплаченные заказы

Рабочий вопрос

Нужно быстро показать, сколько заказов реально оплачено, не смешивая их с новыми и отменёнными строками.

Данные

```
CREATE TABLE orders(id INTEGER, status TEXT, total INTEGER);
INSERT INTO orders VALUES
(1, 'paid', 500),
(2, 'new', 700),
(3, 'paid', 300),
(4, 'cancelled', 900);
```

Запрос

```
SELECT COUNT(*) AS paid_orders
FROM orders
WHERE status = 'paid';
```

Ожидаемый результат

```
[[2]]
```

Почему работает

WHERE оставляет только статус paid.
В таблице два оплаченных заказа.
COUNT(*) считает строки после фильтра.

Где легко ошибиться

COUNT(*) без WHERE посчитал бы все строки, а не подтверждённые оплаты.

Самостоятельная проверка

Почему отменённый заказ на 900 не должен попадать в число оплаченных заказов?
Ответ: Его статус cancelled, а фильтр оставляет только строки со статусом paid.

Задача 2. Найти второй по величине чек

Рабочий вопрос

Нужно найти второй максимальный чек без оконных функций, используя сортировку и смещение.

Данные

```
CREATE TABLE orders(id INTEGER, total INTEGER);  
INSERT INTO orders VALUES (1,500), (2,900), (3,300), (4,700);
```

Запрос

```
SELECT total  
FROM orders  
ORDER BY total DESC  
LIMIT 1 OFFSET 1;
```

Ожидаемый результат

```
[[700]]
```

Почему работает

ORDER BY total DESC ставит самый большой чек первым.
OFFSET 1 пропускает чек 900.
LIMIT 1 берёт следующий чек 700.

Где легко ошибиться

Если есть одинаковые суммы, вопрос о втором чеке нужно уточнить: вторая строка или второе различное значение.

Самостоятельная проверка

Почему решение может стать спорным, если в таблице появятся два заказа на 900?

Ответ: Тогда второй строкой может быть снова 900, но вторым различным значением останется 700.

Задача 3. Найти клиентов без заказов

Рабочий вопрос

Нужно вывести клиентов из справочника, у которых нет ни одного заказа в таблице заказов.

Данные

```
CREATE TABLE clients(id INTEGER, name TEXT);
CREATE TABLE orders(id INTEGER, client_id INTEGER);
INSERT INTO clients VALUES (10, 'Анна'), (20, 'Олег'),
(30, 'Нина');
INSERT INTO orders VALUES (1, 10), (2, 10), (3, 30);
```

Запрос

```
SELECT c.id, c.name
FROM clients c
LEFT JOIN orders o ON o.client_id = c.id
WHERE o.id IS NULL;
```

Ожидаемый результат

```
[[20, "Олег"]]
```

Почему работает

LEFT JOIN сохраняет всех клиентов.

Для клиента 20 не найдено заказа.

WHERE o.id IS NULL оставляет только клиентов без совпадений.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.