

Пункт 1

1. Поиск пользователей по email

Поиск по email без индекса может выполняться медленно, особенно при большом количестве пользователей, поэтому создам индекс на столбце email, так как он часто используется для поиска.

```
CREATE INDEX idx_users_email ON users(email);
```

2. Фильтрация товаров по category и price

Фильтрация по category и price без индексов может приводить к полному сканированию таблицы (Seq Scan).

```
-- Составной индекс для ускорения фильтрации по category и price
CREATE INDEX idx_products_category_price ON products(category, price);

-- Если часто фильтруют только по category:
CREATE INDEX idx_products_category ON products(category);

-- Если часто фильтруют только по price:
CREATE INDEX idx_products_price ON products(price);
```

Пункт 2

1 запрос: Выведите email пользователей и названия товаров, которые они заказывали в 2024 году. Включите только заказы с количеством товаров > 2. Поля результата: user_email, product_title, order_date, quantity

```
SELECT u.email AS user_email,
       p.title AS products_title,
       o.order_date,
       o.quantity
FROM orders AS o
JOIN products AS p ON o.product_id=p.id
JOIN users AS u ON u.id=o.user_id
WHERE o.quantity>2 AND
       extract(YEAR FROM o.order_date)=2024
```

s(+)				
CT u.email AS user_email, p.title AS products_title, o.order_date, o.quantity				
	A-z user_email	A-z products_title	order_date	quantity
	e.cartman@hse.ru	Футболка хлопковая	2024-03-15	3

2 запрос: для каждого пользователя выведите общее количество заказанных и возвращенных им товаров

```
SELECT
    u.id,
    SUM(CASE WHEN o.quantity > 0 THEN o.quantity ELSE 0 END) AS received_products,
    ABS(SUM(CASE WHEN o.quantity < 0 THEN o.quantity ELSE 0 END)) AS returned_products
FROM
    users u
LEFT JOIN
    orders o ON u.id = o.user_id
GROUP BY
    u.id
ORDER BY
    u.id;
```

rs 1	id	received_products	returned_products
	1	3	2
	2	1	0
	3	0	0

3 запрос: выведите товары, которые ЛИБО никогда не заказывались, ЛИБО были заказаны более 3 раз

```
SELECT
    p.id AS product_id,
    p.title,
    CASE WHEN o.quantity > 3 THEN 'Не заказан' ELSE 'Много заказов' END AS status
FROM
    products AS p
JOIN
    orders AS o ON o.product_id = p.id
WHERE o.quantity > 3 OR
    o.quantity = 0
```

products 1	product_id	title	status
	123	product_id	A-Z title A-Z status

Запрос 4: Для каждого пользователя выведите общее количество его заказов и самый дорогой товар, который он когда-либо заказывал.

```
SELECT u.name AS user_name,  
count(o.id) AS total_orders,  
max(p.title) AS most_expensive_product_title,  
max(p.price) AS most_expensive_product_price  
FROM users AS u  
JOIN orders AS o ON o.user_id=u.id  
JOIN products AS p ON p.id=o.product_id  
GROUP BY u.name
```

Введите SQL выражение чтобы отфильтровать результаты				
	A-Z user_name	123 total_orders	A-Z most_expensive_prod	123 most_expensive_prod
	Эрик Картман	2	Футболка хлопковая	1 499,5
	Стэн Марш	1	Ноутбук Pro Extra 15	120 000