

```
/*
```

```
問題1
```

各ユーザーが注文した全商品名と個数を表示してください。

```
*/
```

```
select u.name, p.name, o.quantity from orders o
join users u on o.user_id=u.id
join products p on o.product_id=p.id;
```

```
/*
```

```
問題1'(私が考えた問題)
```

各ユーザーが注文した全商品名と個数を表示してください(同じユーザーが2行以上並ぶ場合は、2行目以降を"にしたい)。

```
*/
```

```
-- このwith句はCTEというこの一回こっきりの問い合わせでのみ使える保存されないビュー
-- を作るやり方
```

```
-- そして、lagはoverで指定されたものでデータを並べ替えた際の直前のものを取るためのwindow関数
```

```
with ordered_products as (
select u.name as user_name, lag(u.name) over (order by u.name) as
prev_user_name, p.name as product_name, o.quantity from orders o
join users u on o.user_id=u.id
join products p on o.product_id=p.id
),
```

```
-- そして、with句の2行目以降はwithを使う必要はない
```

```
-- さらに、ここで問い合わせしなかった列は、次のwith句の外のメインクエリでも問い合わせできなくなるので注意
```

```
-- そして、with句の最後のものの終わりには、をつける必要がない
```

```
adjusted_ordered_products as (
select user_name, prev_user_name,
case
    when user_name=prev_user_name then ''
    else user_name
end as display_user_name,
product_name, quantity from ordered_products
)
select display_user_name, product_name, quantity from
adjusted_ordered_products;
```

```
/*
```

```
問題1''(私が考えた問題)
```

各ユーザーが注文した全商品名と個数を表示してください(同じユーザーが2行以上並ぶ場合は、2行目以降を空欄にし、そこではlagを使わない)。

```
*/
```

```
with ordered_products as (
    select u.id as user_id, u.name as user_name, p.name as
product_name, o.quantity, row_number() over (partition by u.id order
by o.id) as rn from orders o
    join users u on o.user_id = u.id
```

```
    join products p on o.product_id = p.id
)
select
  case
    when rn = 1 then user_name
    else ''
  end as user_name, product_name, quantity from ordered_products;
```