You're working for a company that sells motorcycle parts, and they've asked for some help in analyzing their sales data!

They operate three warehouses in the area, selling both retail and wholesale. They offer a variety of parts and accept credit cards, cash, and bank transfer as payment methods. However, each payment type incurs a different fee.

The board of directors wants to gain a better understanding of wholesale revenue by product line, and how this varies month-to-month and across warehouses. You have been tasked with calculating net revenue for each product line and grouping results by month and warehouse. The results should be filtered so that only `"Wholesale"` orders are included.

They have provided you with access to their database, which contains the following table called `sales`:

Sales

Column	Data type	Description
`order_number`	`VARCHAR`	Unique order number.
`date`	`DATE`	Date of the order, from June to August 2021.
`warehouse`	`VARCHAR`	The warehouse that the order was made from— `North`, `Central`, or `West`.
`client_type`	`VARCHAR`	Whether the order was `Retail` or `Wholesale`.
`product_line`	`VARCHAR`	Type of product ordered.
`quantity`	`INT`	Number of products ordered.
`unit_price`	`FLOAT`	Price per product (dollars).
`total`	`FLOAT`	Total price of the order (dollars).
`payment`	`VARCHAR`	Payment method— `Credit card`, `Transfer`, or `Cash`.
`payment_fee`	`FLOAT`	Percentage of `total` charged as a result of the `payment` method.

Your query output should be presented in the following format:

`product_line`	`month`	`warehouse`	`net_revenue`
product_one	---	---	---
product_one	---	---	---
product_one	---	---	---
product_one	---	---	---
product_one	---	---	---
product_one	---	---	---
product_two	---	---	---
...	...	...	...

Projects Data DataFrame as `revenue_by_product_line`

```
SELECT
        product_line,
        INITCAP(to_char(date, 'month')) AS month,
        warehouse,
```

```
        (SUM(total) - SUM(payment_fee)) AS net_revenue
FROM sales
WHERE client_type = 'Wholesale'
GROUP BY product_line, month, warehouse
ORDER BY product_line ASC, month ASC, net_revenue DESC;
```

in...	product_line	month	wareh...	net_revenue
0	Braking system	August	Central	
1	Braking system	August	West	
2	Braking system	August	North	
3	Braking system	July	Central	
4	Braking system	July	West	
5	Braking system	July	North	
6	Braking system	June	Central	
7	Braking system	June	North	
8	Braking system	June	West	
9	Electrical system	August	North	
10	Electrical system	August	Central	
11	Electrical system	August	West	
12	Electrical system	July	Central	
13	Electrical system	July	North	
14	Electrical system	July	West	
15	Electrical system	June	Central	

Rows: 48

Projects Data DataFrame as `df`

```
SELECT to_char(date, 'month')
FROM sales
LIMIT 5;
```

index	to_char
0	june
1	june
2	june
3	june
4	june

Rows: 5