Revving Up Insights: Analyzing Motorcycle Part Sales with SQL

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`:

Revving Up Insights: Analyzing Motorcycle Part Sales with SQL

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 revenu

```
SELECT product_line,
    CASE WHEN EXTRACT('month' from date) = 6 THEN 'June'
        WHEN EXTRACT('month' from date) = 7 THEN 'July'
        WHEN EXTRACT('month' from date) = 8 THEN 'August'
    END as month,
    warehouse,
    SUM(total) - SUM(payment_fee) AS net_revenue
FROM sales
WHERE client_type = 'Wholesale'
GROUP BY product_line, warehouse, month
ORDER BY product_line, month, net_revenue DESC
```

index	product_line	month	warehouse	net_revenue
0	Braking system	August	Central	3039
1	Braking system	August	West	2500
2	Braking system	August	North	1770
3	Braking system	July	Central	3778
4	Braking system	July	West	3060
5	Braking system	July	North	2594
6	Braking system	June	Central	3684
7	Braking system	June	North	1487
8	Braking system	June	West	1212
9	Electrical system	August	North	4721
10	Electrical system	August	Central	3126
11	Electrical system	August	West	1241
12	Electrical system	July	Central	5577
13	Electrical system	July	North	1710
14	Electrical system	July	West	449
15	Electrical system	June	Central	2904

Rows: 48   Expand

Extended Project below

The finance team is exploring ways to reduce transaction costs and improve profitability. They've asked you to determine the most profitable payment method for each warehouse in each month. Calculate the net revenue for each payment method, grouped by warehouse and month, and identify the top payment method for each combination.

Revving Up Insights: Analyzing Motorcycle Part Sales with SQL

Projects Data DataFrame as df2

```
WITH monthly_revenue AS (
    SELECT
        warehouse,
        EXTRACT(MONTH FROM date) AS month,
        payment,
        SUM(total - payment_fee) AS net_revenue
    FROM public.sales
    GROUP BY warehouse, month, payment
),
ranked_methods AS (
    SELECT *,
           RANK() OVER (PARTITION BY warehouse, month ORDER BY net_revenue DESC) AS payment_rank
    FROM monthly_revenue
)
SELECT
    warehouse,
    month,
    payment,
    net_revenue
FROM ranked_methods
WHERE payment_rank = 1
ORDER BY warehouse, month;
```

	w			net…
0	Central	6	Transfer	23453.08
1	Central	7	Transfer	23893.59
2	Central	8	Transfer	31509
3	North	6	Transfer	17000.12
4	North	7	Transfer	17585.25
5	North	8	Transfer	23480.13
6	West	6	Transfer	8645.98
7	West	7	Transfer	7606.51
8	West	8	Transfer	6466.42

Rows: 9 Expand

The marketing team is planning a targeted campaign and wants to know the most popular product lines for retail and wholesale customers.

They have given you the task to find the top 3 most ordered product lines for each client type.

Projects Data DataFrame as d

```
WITH Client AS (
    SELECT
        client_type,
        product_line,
        COUNT(*) AS product_line_count,
        RANK() OVER (PARTITION BY client_type ORDER BY COUNT(*) DESC) AS rank
    FROM public.sales
    GROUP BY client_type, product_line
)
SELECT
    client_type,
    product_line,
    product_line_count
FROM Client
WHERE rank <= 3
ORDER BY client_type, rank;
```

	clie…	product_line	product_line_c…
0	Retail	Suspension & traction	177
1	Retail	Braking system	175
2	Retail	Electrical system	155
3	Wholesale	Braking system	55
4	Wholesale	Suspension & traction	51
5	Wholesale	Frame & body	38
6	Wholesale	Electrical system	38

Rows: 7 Expand