Welcome! You are now in DataLab.

You successfully completed your project and are looking for some additional related challenges. This DataLab workbook contains the official solution from our curriculum staff, along with Additional Challenges at the bottom. If you would like a quick overview of DataLab, please refer to the help menu. You can easily share your project with your friends and colleagues when you're done.

Good luck with your additional challenges!

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

```sql
-- Start coding here
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	wareho
0	Braking system	August	Cent
1	Braking system	August	West
2	Braking system	August	Nort
3	Braking system	July	Cent
4	Braking system	July	West
5	Braking system	July	Nort
6	Braking system	June	Cent
7	Braking system	June	Nort
8	Braking system	June	West
9	Electrical system	August	Nort
10	Electrical system	August	Cent
11	Electrical system	August	West
12	Electrical system	July	Cent
13	Electrical system	July	Nort
14	Electrical system	July	West
15	Electrical system	June	Cent

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.

Projects Data DataFrame as df

```
WITH ranked_sales AS (
  SELECT
    warehouse,
    EXTRACT(MONTH FROM date) as month,
    payment,
    SUM(total) - SUM(payment_fee) AS net_revenue,
    RANK() OVER (
        PARTITION BY warehouse, EXTRACT(MONTH FROM date)
        ORDER BY SUM(total) - SUM(payment_fee) DESC
    ) AS rank
  FROM sales
  GROUP BY warehouse, month, payment
)

SELECT
    warehouse,
    CASE
        WHEN month = 6 THEN 'June'
        WHEN month = 7 THEN 'July'
        WHEN month = 8 THEN 'August'
    END AS month,
    payment,
    net_revenue
FROM ranked_sales
WHERE rank = 1
ORDER BY warehouse, month DESC;
```

	w			net...
0	Central	June	Transfer	23453.08
1	Central	July	Transfer	23893.59
2	Central	August	Transfer	31509
3	North	June	Transfer	17000.12
4	North	July	Transfer	17585.25
5	North	August	Transfer	23480.13
6	West	June	Transfer	8645.98
7	West	July	Transfer	7606.51
8	West	August	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 ranked AS (
SELECT
    client_type,
    product_line,
    SUM(quantity) AS num_of_orders,
    RANK() OVER (
        PARTITION BY client_type
        ORDER BY SUM(quantity) DESC
    ) AS rank
FROM sales
GROUP BY client_type, product_line
)

SELECT
    client_type,
    product_line,
    num_of_orders
FROM ranked
WHERE rank <= 3
```

	clie...	product_line	num_o...
0	Retail	Suspension & traction	1017
1	Retail	Electrical system	862
2	Retail	Braking system	862
3	Wholesale	Braking system	1268
4	Wholesale	Suspension & traction	1128
5	Wholesale	Frame & body	932

Rows: 6

Expand