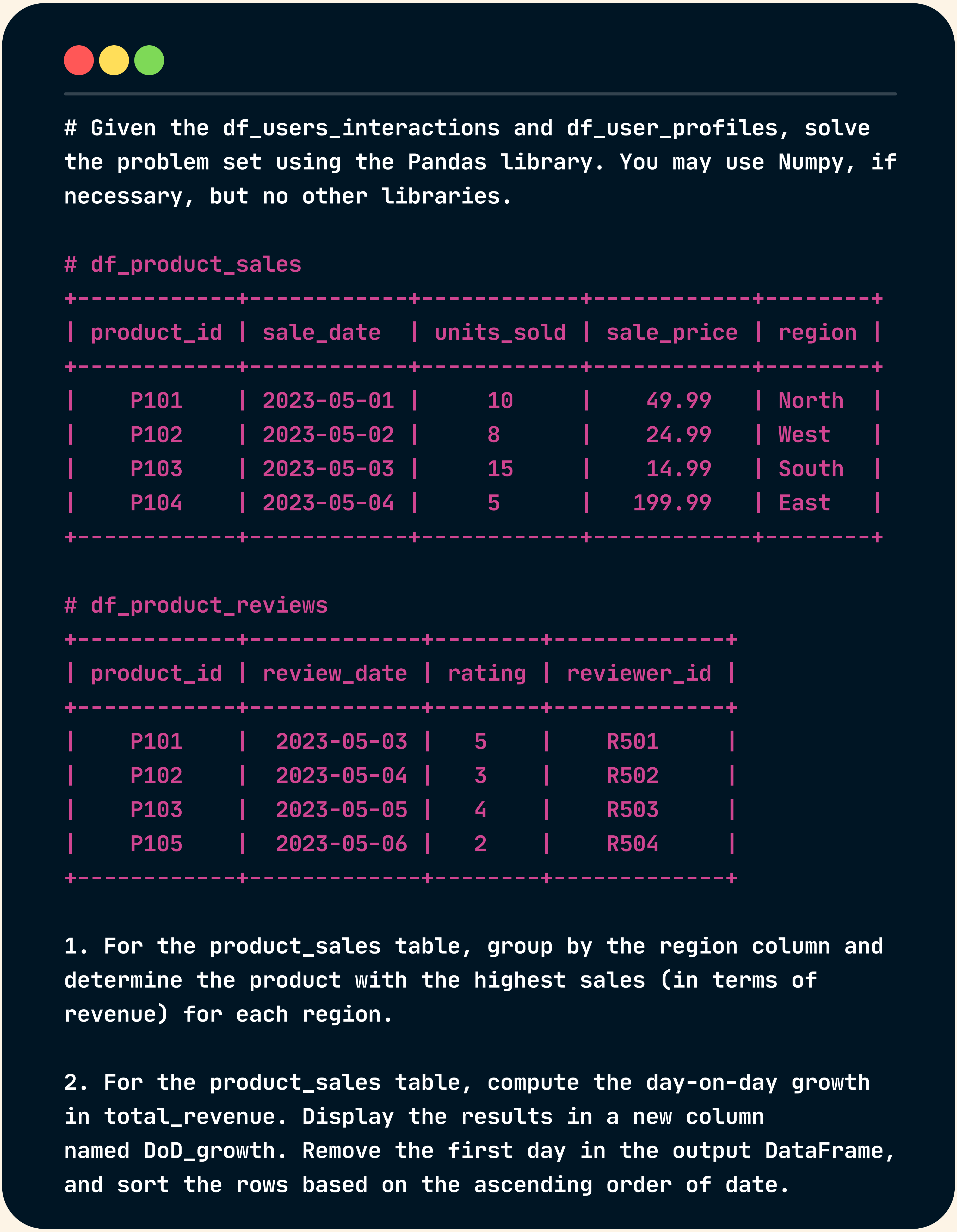
DataInterview.com

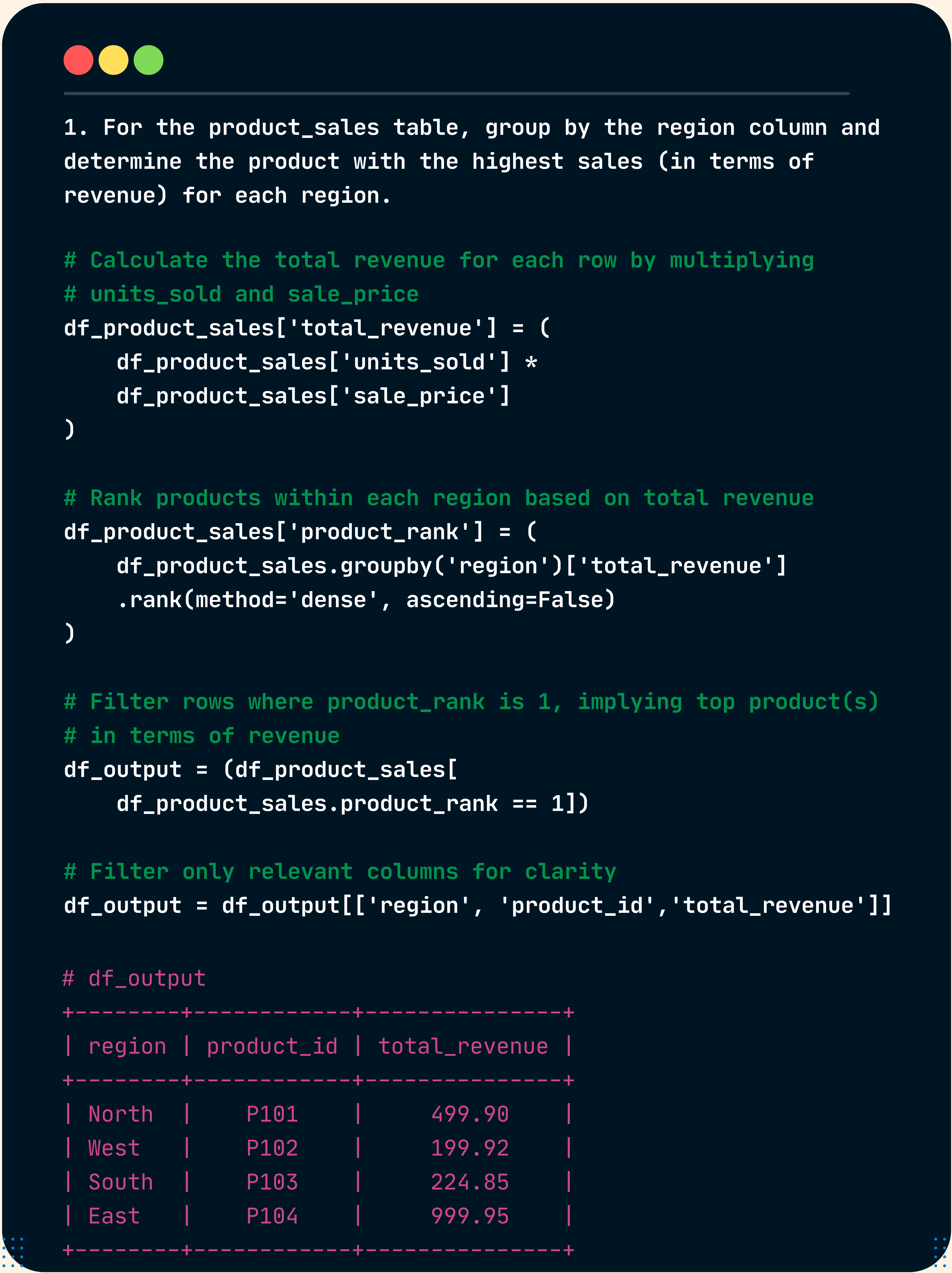
Python Pandas Interview Questions



```
# Given the df_users_interactions and df_user_profiles, solve
the problem set using the Pandas library. You may use Numpy, if
necessary, but no other libraries.

# df_product_sales
+------------+------------+------------+------------+--------+
| product_id | sale_date  | units_sold | sale_price | region |
+------------+------------+------------+------------+--------+
|    P101    | 2023-05-01 |     10     |    49.99   | North  |
|    P102    | 2023-05-02 |     8      |    24.99   | West   |
|    P103    | 2023-05-03 |     15     |    14.99   | South  |
|    P104    | 2023-05-04 |     5      |   199.99   | East   |
+------------+------------+------------+------------+--------+

# df_product_reviews
+------------+-------------+--------+-------------+
| product_id | review_date | rating | reviewer_id |
+------------+-------------+--------+-------------+
|    P101    |  2023-05-03 |   5    |    R501     |
|    P102    |  2023-05-04 |   3    |    R502     |
|    P103    |  2023-05-05 |   4    |    R503     |
|    P105    |  2023-05-06 |   2    |    R504     |
+------------+-------------+--------+-------------+

1. For the product_sales table, group by the region column and
determine the product with the highest sales (in terms of
revenue) for each region.

2. For the product_sales table, compute the day-on-day growth
in total_revenue. Display the results in a new column
named DoD_growth. Remove the first day in the output DataFrame,
and sort the rows based on the ascending order of date.
```

amazon Solutions #1

DataInterview.com

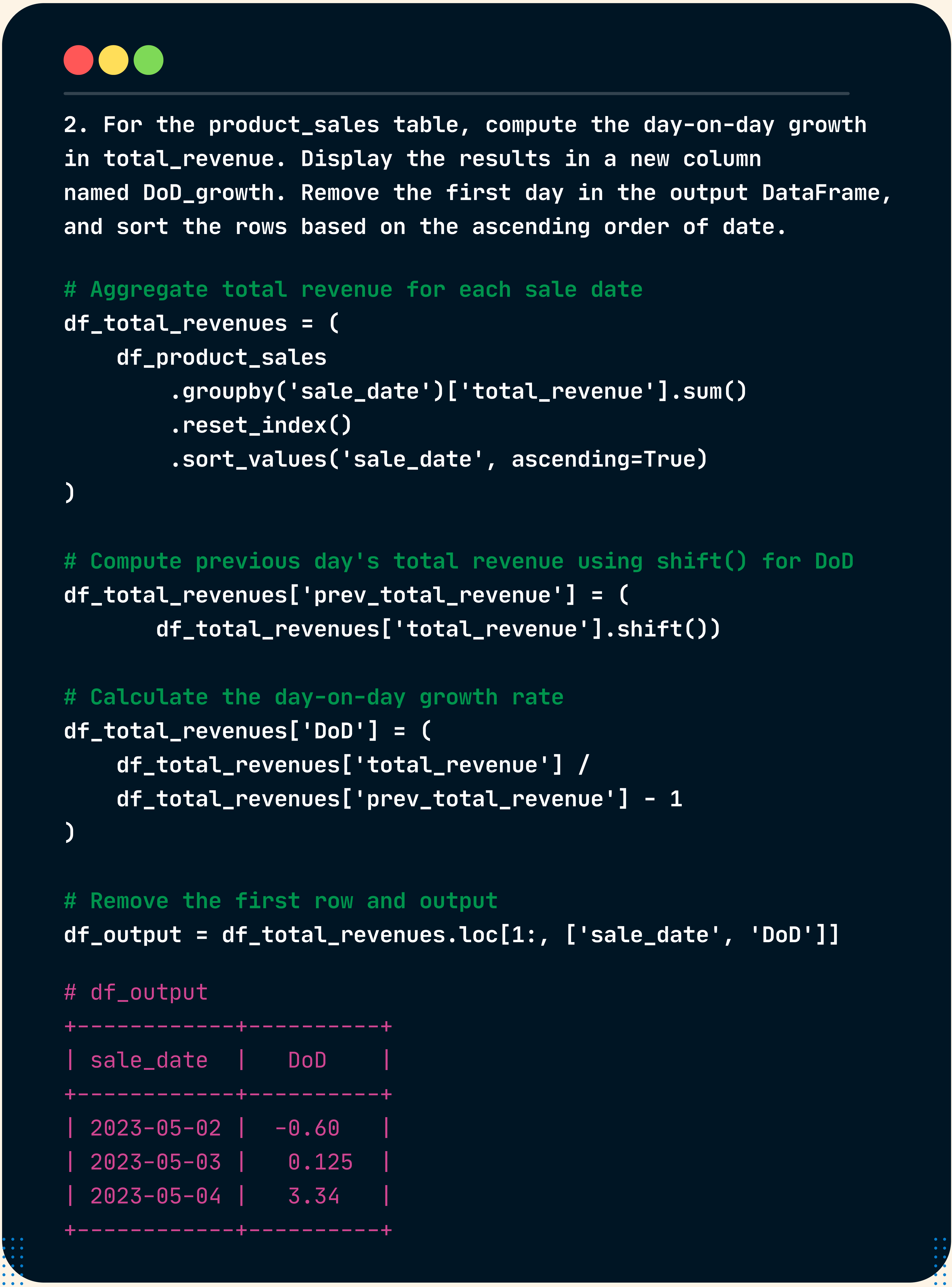


1. For the product_sales table, group by the region column and determine the product with the highest sales (in terms of revenue) for each region.

```
# Calculate the total revenue for each row by multiplying
# units_sold and sale_price
df_product_sales['total_revenue'] = (
    df_product_sales['units_sold'] *
    df_product_sales['sale_price']
)

# Rank products within each region based on total revenue
df_product_sales['product_rank'] = (
    df_product_sales.groupby('region')['total_revenue']
    .rank(method='dense', ascending=False)
)

# Filter rows where product_rank is 1, implying top product(s)
# in terms of revenue
df_output = (df_product_sales[
    df_product_sales.product_rank == 1])

# Filter only relevant columns for clarity
df_output = df_output[['region', 'product_id','total_revenue']]

# df_output
+--------+------------+---------------+
| region | product_id | total_revenue |
+--------+------------+---------------+
| North  |    P101    |     499.90    |
| West   |    P102    |     199.92    |
| South  |    P103    |     224.85    |
| East   |    P104    |     999.95    |
+--------+------------+---------------+
```

amazon Solutions #2

DataInterview.com



2. For the product_sales table, compute the day-on-day growth in total_revenue. Display the results in a new column named DoD_growth. Remove the first day in the output DataFrame, and sort the rows based on the ascending order of date.

```
# Aggregate total revenue for each sale date
df_total_revenues = (
    df_product_sales
        .groupby('sale_date')['total_revenue'].sum()
        .reset_index()
        .sort_values('sale_date', ascending=True)
)

# Compute previous day's total revenue using shift() for DoD
df_total_revenues['prev_total_revenue'] = (
       df_total_revenues['total_revenue'].shift())

# Calculate the day-on-day growth rate
df_total_revenues['DoD'] = (
    df_total_revenues['total_revenue'] /
    df_total_revenues['prev_total_revenue'] - 1
)

# Remove the first row and output
df_output = df_total_revenues.loc[1:, ['sale_date', 'DoD']]

# df_output
+------------+----------+
| sale_date  |   DoD    |
+------------+----------+
| 2023-05-02 |  -0.60   |
| 2023-05-03 |   0.125  |
| 2023-05-04 |   3.34   |
+------------+----------+
```

DataInterview.com

Join DataInterview.com

Access courses, mock interviewing and private community with coaches and candidates like you

Created by interviewers at top companies like Google and Meta