

# HeatTransformers Assignment

## For Data Engineer

The assignment is split into two parts:

- Showcase Python development skills
- Showcase PySpark/SQL skills

The topics are rather straightforward and the assignment is to be completed subsequently as there is a dependency of the SQL assignment on the Python assignment. You should estimate the assignment to take a maximum of 4 hours. We expect you to track and manage your time properly. Please read both assignments carefully before starting, you might be able to re-use some of your code.

## Do's

- ✓ Use libraries (at the very least PySpark and its dependencies).
- ✓ Document your code, appropriate to the assumption others will maintain it.
- ✓ Make sure we can run your code locally. Send us everything we need.
- ✓ Scroll down to "input data" to have a look at the dummy data.

## Deliverables

- ✓ The scripts you wrote to complete the assignments, which we can run locally.
- ✓ Separate documentation you feel is necessary or supportive to your implementation or how we can properly run your code.
- ✓ Any other dependencies/packages/shell scripts you have used to complete this.

## Python

The objective is to write code that facilitates the process of shipping products between different warehouse units on a daily basis for 10 days. Your script should track which product is stored in which warehouse unit on what day and either print this to the console or write it to a file.

The warehouse units contain the products that need to be shipped and follow certain formulas to determine the next warehouse the product needs to be shipped to.

- Products is a list of product numbers, which are always integers.
- The product number in combination with the formula of the warehouse it is stored in determines the shipping location of the product.
- When a product is shipped, its number changes based on the formula of the warehouse it was sent from, rounded up.
- Before a product is shipped, its number is changed based on the following formula:

$$\text{New number} = \text{Old number} / 3$$

- A product is shipped to a new Warehouse Unit every day, resulting in a completely new inventory every day.

This is an example of how it works based on the example and warehouse unit 0 from the data:

Unset

Warehouse Unit 0:

Warehouse Unit sends a product with the number 74.

The product number is multiplied with 7, which results in 518.

The product number is divided by 3 based on the abovementioned formula, resulting in 173, rounded up.

The current product number 173 is not divisible by 5.

The product is subsequently sent to Warehouse Unit 6

An example of what we want you to print/write:

Unset

Day 1:

Warehouse Unit 0: 20, 23, 27, 26

Warehouse Unit 1: 2080, 25, 167, 207, 401, 1046

Warehouse Unit 2:

Warehouse Unit 3:

Day 2,

Warehouse Unit 0: 695, 10, 71, 135, 350

Warehouse Unit 1: 43, 49, 58, 55, 362

Warehouse Unit 2:

Warehouse Unit 3:

...

Day 10,

Warehouse Unit 0: 91, 16, 20, 98

Warehouse Unit 1: 481, 245, 22, 26, 1092, 30

Warehouse Unit 2:

Warehouse Unit 3:

## PySpark/SQL

The objective here is to deliver one dataset. The dataset is a combination of the data you started with in the Python assignment. You are free to complete this assignment in either PySpark or SQL (meaning we want to either see a .py or a .sql file). This dataset has to be

optimized for reporting and analysis where we create graphs in which we can still “slice” the data.

The dataset should at least contain the following variables with their respective data types. We will leave it to you to structure the dataset as long as it will allow us to determine:

- What the product’s initial product number was.
- Where the product was stored on which day during the 10 days.
- What the product number was on which day during the days.

Variable	Data Type	Description
warehouse_id	integer	Identification number of the warehouse unit the products are stored in.
warehouse_name	string	Name of the warehouse unit related to the warehouse id.
warehouse_formula	string	Formula belonging to the warehouse to change the product number.
product_origin	integer	Identification number of the warehouse unit the product is currently stored in and sent from.
product_destination	integer	Identification number of the warehouse unit the product is sent to.
starting_product_id	integer	Starting identification number of a product in the ecosystem.
current_product_id	integer	Current identification number of a product in the ecosystem based on the amount of days passed.
test_bool	boolean	Boolean which shows whether the test belonging to the specific warehouse failed or succeeded.
day_number	integer	Number of the day which shows the amount of iterations of shipping have past (range 1 to 10).

# Input data:

## Warehouse Unit 0:

Starting Products: 74, 64, 74, 63, 53

Warehouse Formula:  $\text{New} = \text{Old} * 7$

Test: divisible by 5

If true: ship product to Warehouse Unit 1

If false: ship product to Warehouse Unit 6

## Warehouse Unit 1:

Starting Products: 69, 99, 95, 62

Warehouse Formula:  $\text{New} = \text{Old} * \text{Old}$

Test: divisible by 17

If true: ship product to Warehouse Unit 2

If false: ship product to Warehouse Unit 5

## Warehouse Unit 2:

Starting Products: 59, 81

Warehouse Formula:  $\text{New} = \text{Old} + 8$

Test: divisible by 7

If true: ship product to Warehouse Unit 4

If false: ship product to Warehouse Unit 3

## Warehouse Unit 3:

Starting Products: 50, 67, 63, 57, 63, 83, 97

Warehouse Formula:  $\text{New} = \text{Old} + 4$

Test: divisible by 13

If true: ship product to Warehouse Unit 0

If false: ship product to Warehouse Unit 7

## Warehouse Unit 4:

Starting Products: 61, 94, 85, 52, 81, 90, 94, 70

Warehouse Formula:  $\text{New} = \text{Old} + 3$

Test: divisible by 19

If true: ship product to Warehouse Unit 7

If false: ship product to Warehouse Unit 3

## Warehouse Unit 5:

Starting Products: 69

Warehouse Formula:  $\text{New} = \text{Old} + 5$

Test: divisible by 3

If true: ship product to Warehouse Unit 4

If false: ship product to Warehouse Unit 2

## Warehouse Unit 6:

Starting Products: 54, 55, 58

Warehouse Formula:  $\text{New} = \text{Old} + 7$

Test: divisible by 11

If true: ship product to Warehouse Unit 1

If false: ship product to Warehouse Unit 5

Warehouse Unit 7:

Starting Products: 79, 51, 83, 88, 93, 76

Warehouse Formula:  $\text{New} = \text{Old} * 3$

Test: divisible by 2

If true: ship product to Warehouse Unit 0

If false: ship product to Warehouse Unit 6

# The .txt-file contents

Warehouse Unit: 0

Starting Products: 74, 64, 74, 63, 53

Warehouse Formula:  $\text{New} = \text{Old} * 7$

Test: / 5

True: ship to Warehouse Unit 1

False: ship to Warehouse Unit 6

Warehouse Unit: 1

Starting Products: 69, 99, 95, 62

Warehouse Formula:  $\text{New} = \text{Old} * \text{Old}$

Test: / 17

True: ship to Warehouse Unit 2

False: ship to Warehouse Unit 5

Warehouse Unit: 2

Starting Products: 59, 81

Warehouse Formula:  $\text{New} = \text{Old} + 8$

Test: / 7

True: ship to Warehouse Unit 4

False: ship to Warehouse Unit 3

Warehouse Unit: 3

Starting Products: 50, 67, 63, 57, 63, 83, 97

Warehouse Formula:  $\text{New} = \text{Old} + 4$

Test: / 13

True: ship to Warehouse Unit 0

False: ship to Warehouse Unit 7

Warehouse Unit: 4

Starting Products: 61, 94, 85, 52, 81, 90, 94, 70

Warehouse Formula:  $\text{New} = \text{Old} + 3$

Test: / 19

True: ship to Warehouse Unit 7

False: ship to Warehouse Unit 3

Warehouse Unit: 5

Starting Products: 69

Warehouse Formula:  $\text{New} = \text{Old} + 5$

Test: / 3

True: ship to Warehouse Unit 4

False: ship to Warehouse Unit 2

Warehouse Unit: 6

Starting Products: 54, 55, 58

Warehouse Formula:  $\text{New} = \text{Old} + 7$

Test: / 11

True: ship to Warehouse Unit 1

False: ship to Warehouse Unit 5

Warehouse Unit: 7

Starting Products: 79, 51, 83, 88, 93, 76

Warehouse Formula:  $\text{New} = \text{Old} * 3$

Test: / 2

True: ship to Warehouse Unit 0

False: ship to Warehouse Unit 6