DELIVERY PROCESS ANALYSIS

MÜSLI PROJECT

DATA ANALYTICS DEPARTMENT

April 2024

AGENDA

1 ORDER PROCESS FLOW CHART

2 KPIs

3 OUR PROCESS

4 OUR FINDINGS

5 CONCLUSIONS

AGENDA

1 ORDER PROCESS FLOW CHART

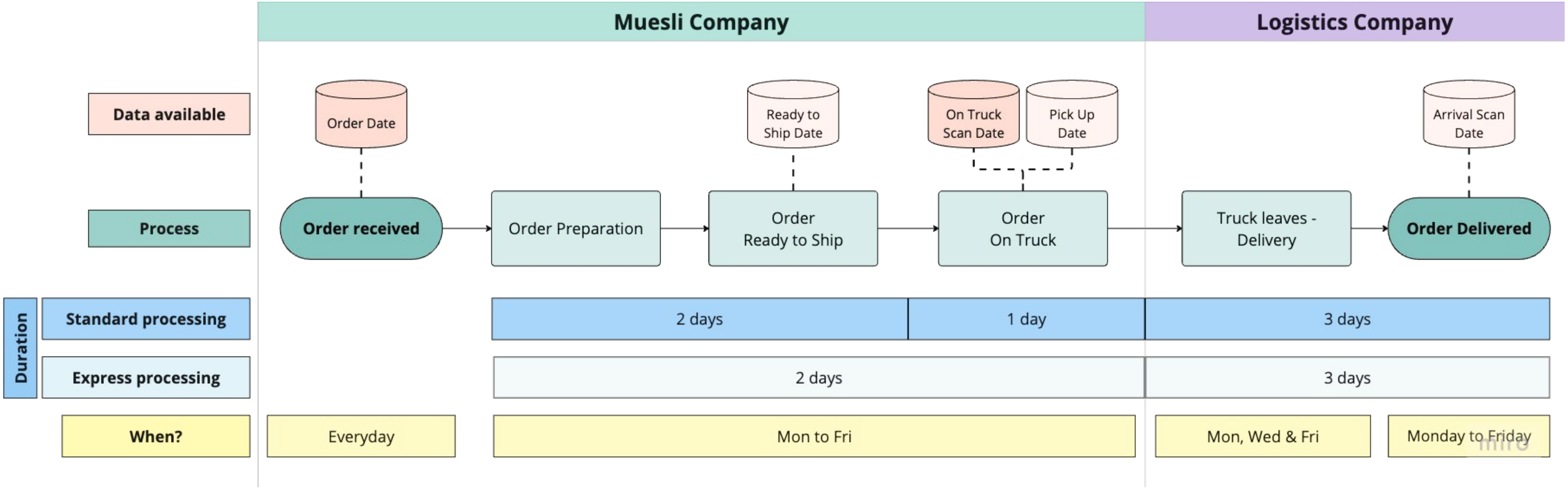
2 KPIs

3 OUR PROCESS

4 OUR FINDINGS

5 CONCLUSIONS

1 ORDER PROCESS FLOW CHART



Muesli Company

Logistics Company

Data available

Order Date

Ready to Ship Date

On Truck Scan Date

Pick Up Date

Arrival Scan Date

Process

Order received

Order Preparation

Order Ready to Ship

Order On Truck

Truck leaves - Delivery

Order Delivered

Duration

Standard processing

2 days

1 day

3 days

Express processing

2 days

3 days

When?

Everyday

Mon to Fri

Mon, Wed & Fri

Monday to Friday

AGENDA

1 ORDER PROCESS FLOW CHART

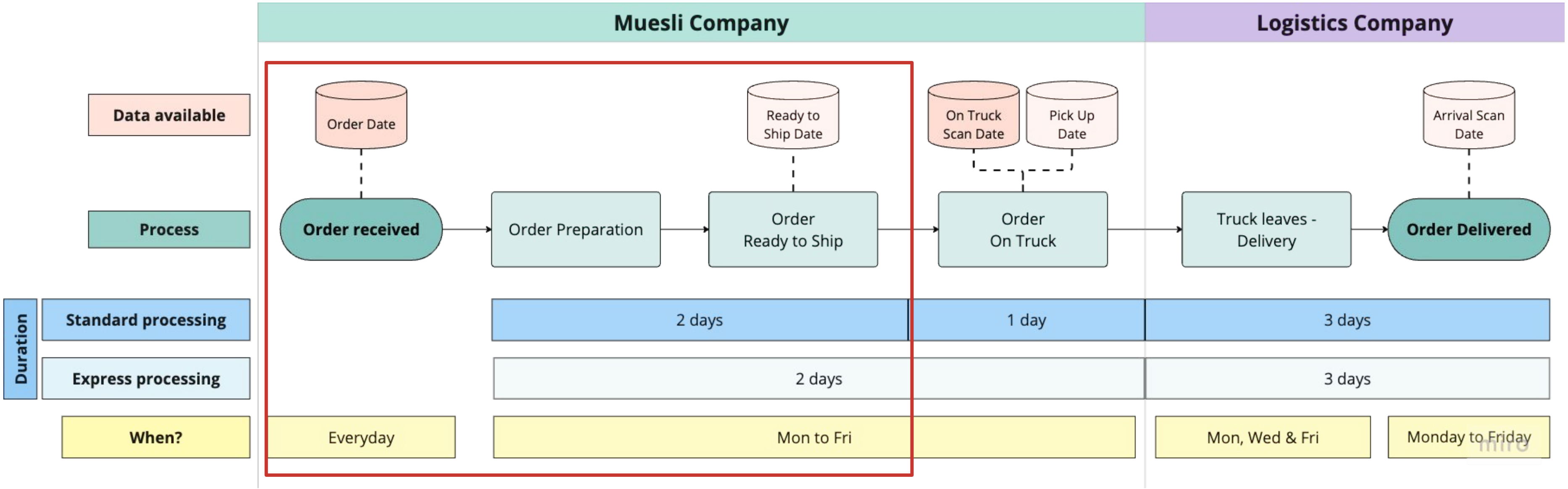
2 KPIs

3 OUR PROCESS

4 OUR FINDINGS

5 CONCLUSIONS

2 KPIs



Muesli Company

Logistics Company

Data available

Order Date

Ready to Ship Date

On Truck Scan Date

Pick Up Date

Arrival Scan Date

Process

Order received

Order Preparation

Order Ready to Ship

Order On Truck

Truck leaves - Delivery

Order Delivered

Duration

Standard processing

2 days

1 day

3 days

Express processing

2 days

3 days

When?

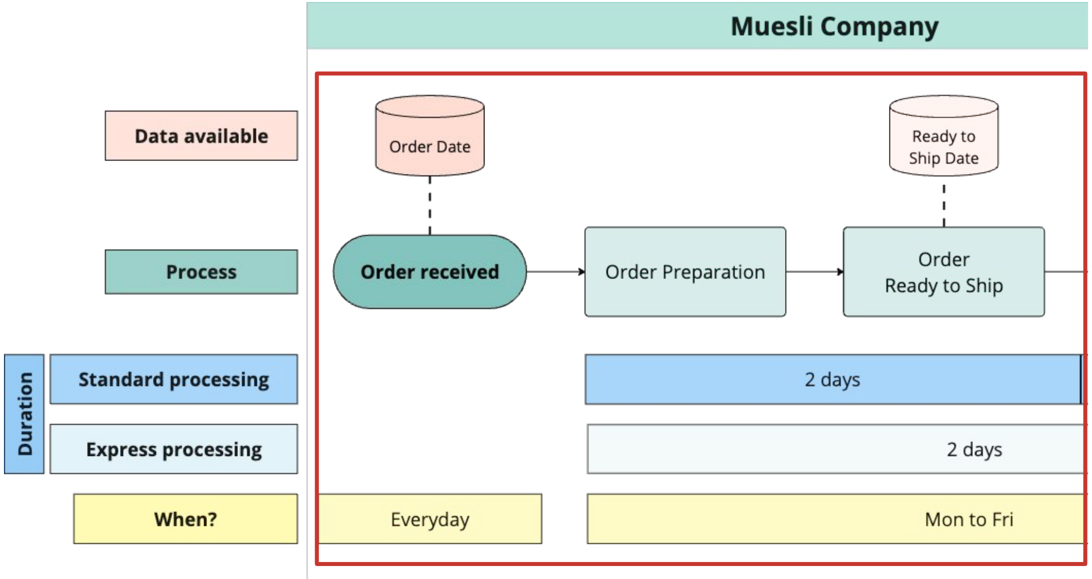
Everyday

Mon to Fri

Mon, Wed & Fri

Monday to Friday

2 KPIs



KPI 1 : Order Fulfillment Time

- Measures time from “Order received” to “Order Ready to Ship” in days
- Calculation:
“Ready to Ship Date” - “Order Date”
- Validate:

  Order preparation time takes 2 days

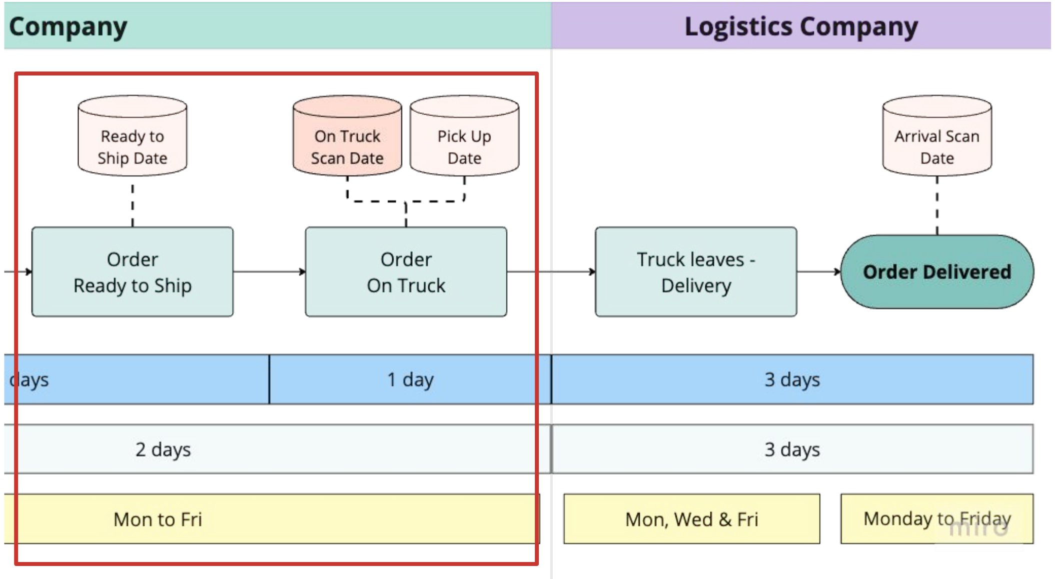
2 KPIs

KPI 2 : Order Loading Time

- Measures time from “Order Ready to Ship” to “On Truck Scan Date” in days

- Calculation:
  “On Truck Scan Date” - “Ready to Ship Date”

- Validate:
  Order loading time takes 1 day for Standard Processing and same day for Express Processing



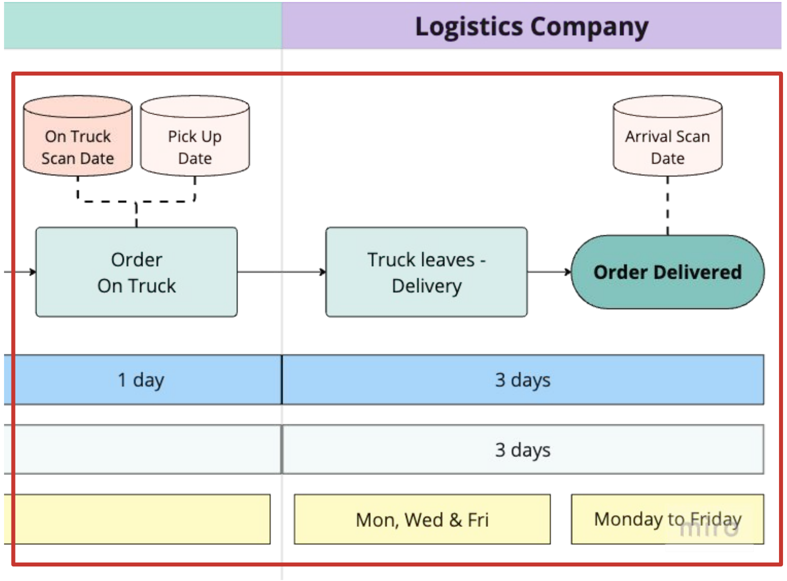
2 KPIs

KPI 3 : Order Delivery Time

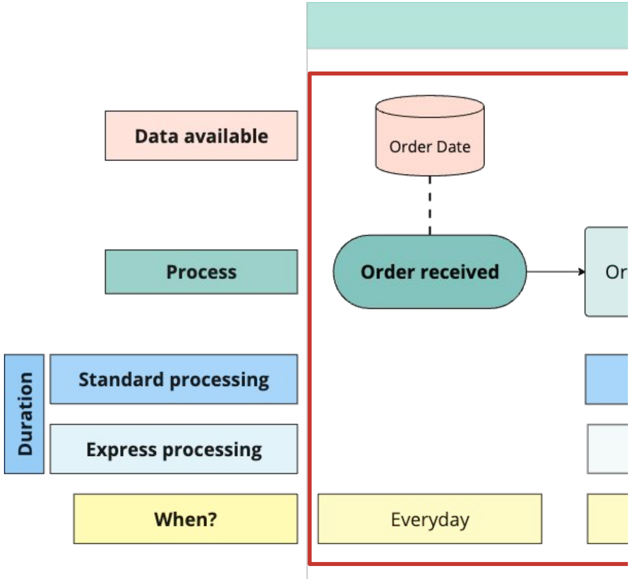
- Measures time from “Order on Truck” to “Order Delivered” in days

- Calculation:
   “Arrival Scan Date” - “On Truck Scan Date”

- Validate:
  Order delivery time takes 3 days as Logistics Company calculated

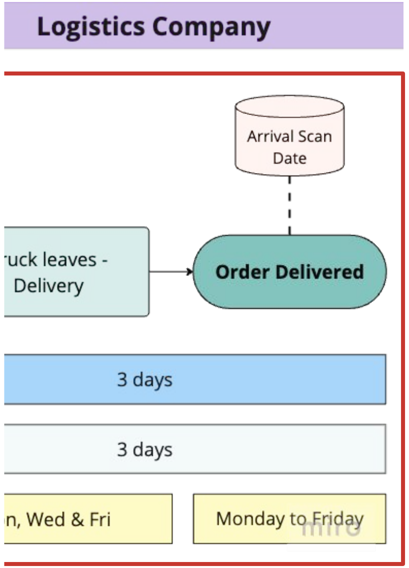


2 KPIs

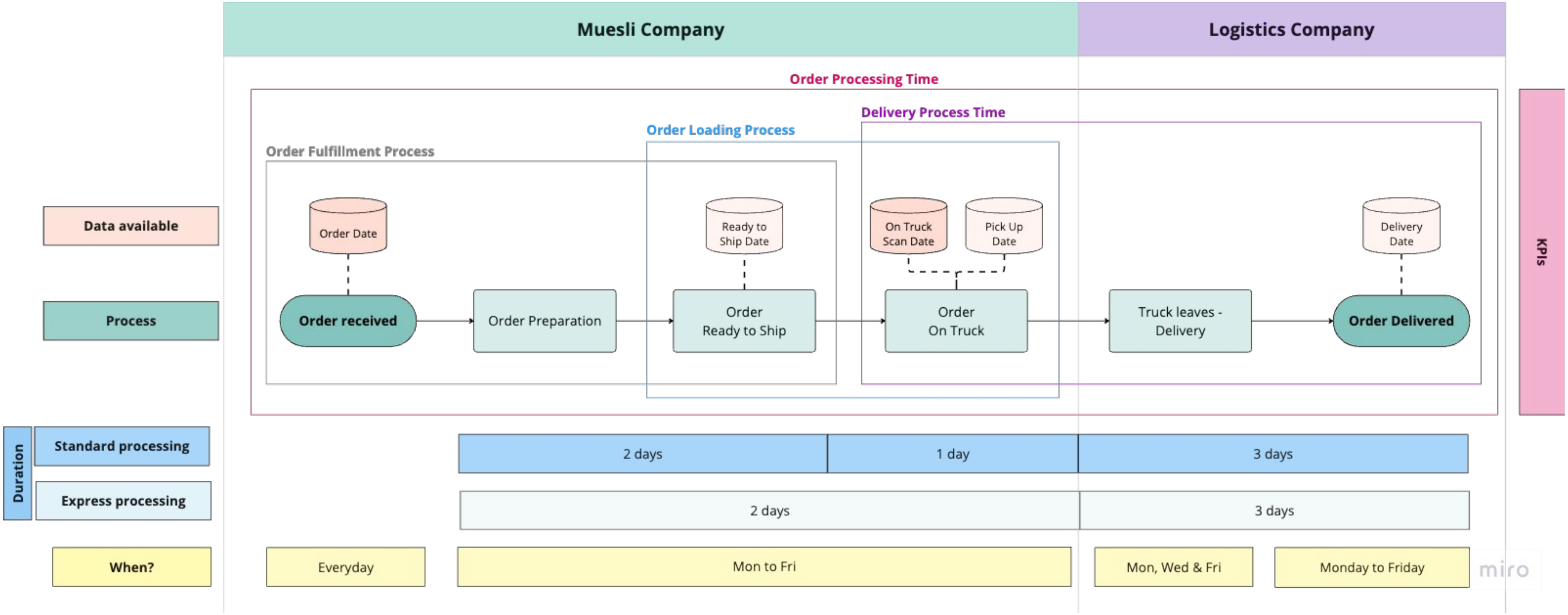


KPI 4 : Order Processing Time

- Measures time from “Order Date” to “ Arrival Scan Date” in days
- Calculation:
  “Arrival Scan Date” - “Order Date”
- Validate:
  Order processing time takes 6 days for standard processing and 5 days for express processing



2 KPIs



Muesli Company

Logistics Company

Order Processing Time

Delivery Process Time

Order Loading Process

Order Fulfillment Process

Data available

Order Date

Ready to Ship Date

On Truck Scan Date

Pick Up Date

Delivery Date

Process

Order received

Order Preparation

Order Ready to Ship

Order On Truck

Truck leaves - Delivery

Order Delivered

KPIs

Duration

Standard processing: 2 days | 1 day | 3 days

Express processing: 2 days | 3 days

When?: Everyday | Mon to Fri | Mon, Wed & Fri | Monday to Friday

AGENDA

1 ORDER PROCESS FLOW CHART

2 KPIs

3 **OUR PROCESS**

4 OUR FINDINGS

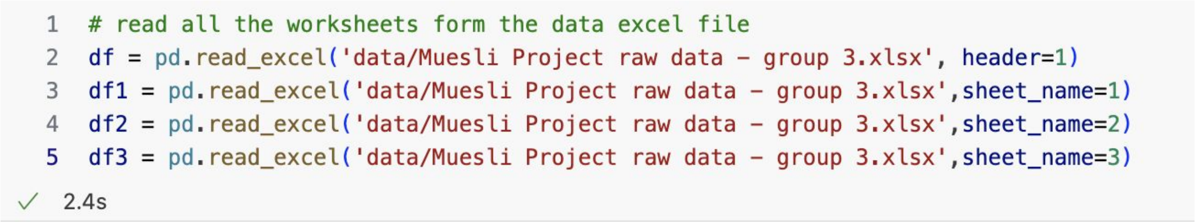
5 CONCLUSIONS

3 OUR PROCESS

- Data examination
- Data cleaning
- Merging data

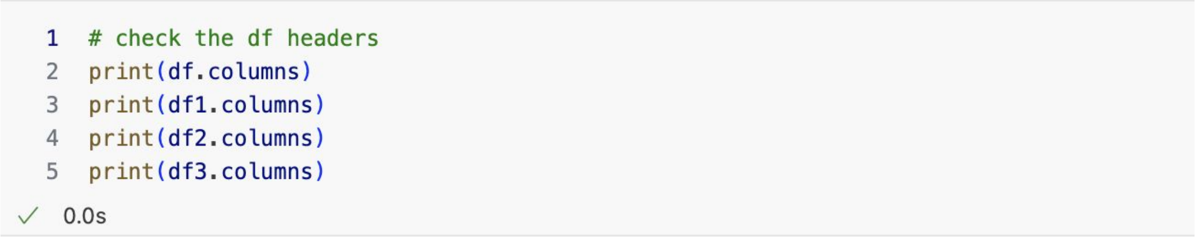
3 OUR PROCESS

Data examination



```python
# read all the worksheets form the data excel file
df = pd.read_excel('data/Muesli Project raw data - group 3.xlsx', header=1)
df1 = pd.read_excel('data/Muesli Project raw data - group 3.xlsx',sheet_name=1)
df2 = pd.read_excel('data/Muesli Project raw data - group 3.xlsx',sheet_name=2)
df3 = pd.read_excel('data/Muesli Project raw data - group 3.xlsx',sheet_name=3)
```

✓ 2.4s



```python
# check the df headers
print(df.columns)
print(df1.columns)
print(df2.columns)
print(df3.columns)
```

✓ 0.0s

3 OUR PROCESS

Data examination

Orders

Index
Order ID
Order Date
Ship Mode
Customer ID
Customer Name
Origin Channel
Country/Region
City
State
Postal Code
Region
Category
Sub-Category
Product ID
Sales
Quantity
Discount
Profit

Campaign Data

Order ID
Arrival Scan Date
Customer Name

Order Process Data

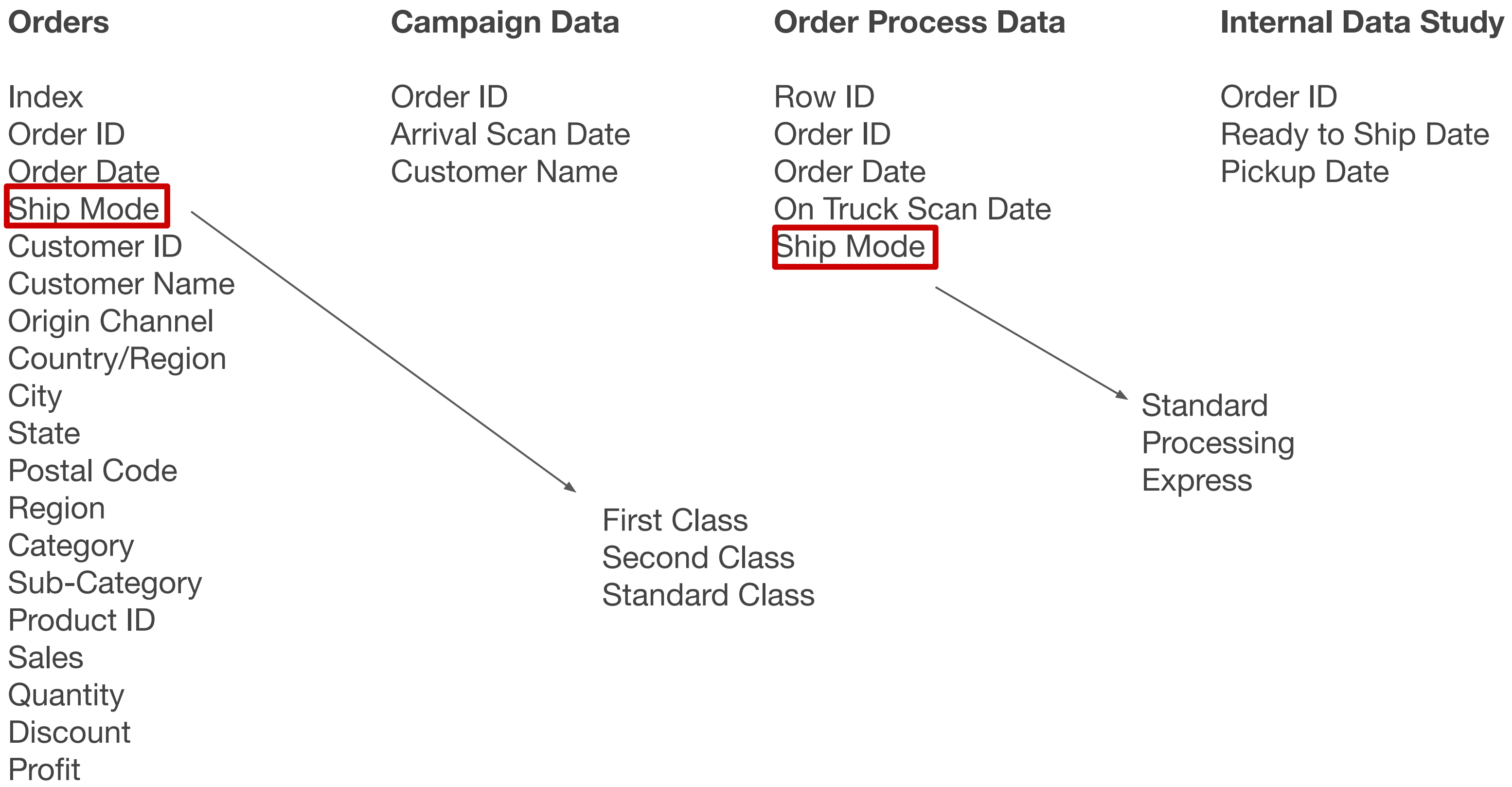
Row ID
Order ID
Order Date
On Truck Scan Date
Ship Mode

Internal Data Study

Order ID
Ready to Ship Date
Pickup Date

3 OUR PROCESS

Data examination



Orders

Index
Order ID
Order Date
Ship Mode
Customer ID
Customer Name
Origin Channel
Country/Region
City
State
Postal Code
Region
Category
Sub-Category
Product ID
Sales
Quantity
Discount
Profit

First Class
Second Class
Standard Class

Campaign Data

Order ID
Arrival Scan Date
Customer Name

Order Process Data

Row ID
Order ID
Order Date
On Truck Scan Date
Ship Mode

Standard
Processing
Express

Internal Data Study

Order ID
Ready to Ship Date
Pickup Date

3 OUR PROCESS

Data examination

Orders

Index
Order ID
Order Date
Ship Mode
Customer ID
Customer Name
Origin Channel
Country/Region
City
State
Postal Code
Region
Category
Sub-Category
Product ID
Sales
Quantity
Discount
Profit

Campaign Data

Order ID
Arrival Scan Date
Customer Name

Order Process Data

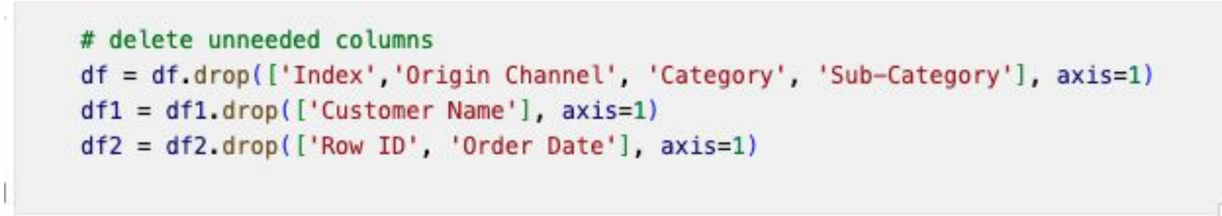
Row ID
Order ID
Order Date
On Truck Scan Date
Ship Mode

Internal Data Study

Order ID
Ready to Ship Date
Pickup Date

3 OUR PROCESS

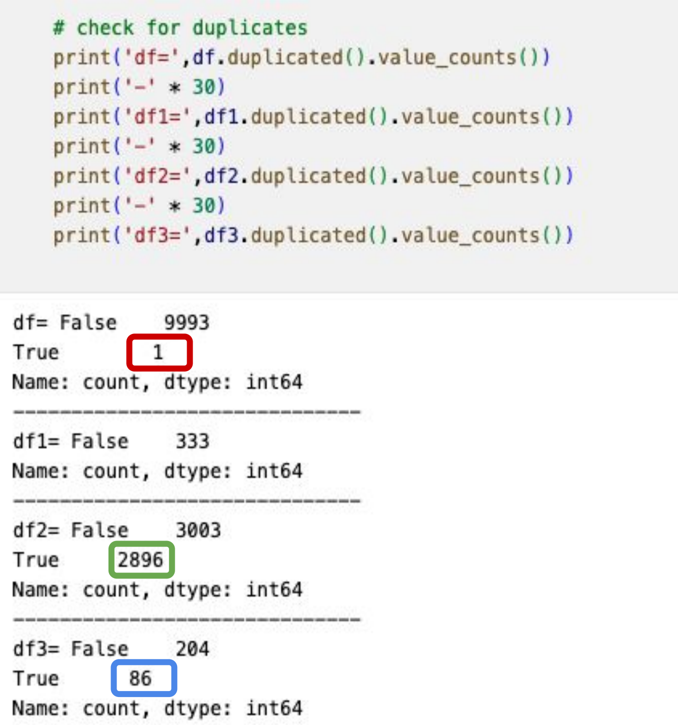
Data cleaning



```python
# delete unneeded columns
df = df.drop(['Index','Origin Channel', 'Category', 'Sub-Category'], axis=1)
df1 = df1.drop(['Customer Name'], axis=1)
df2 = df2.drop(['Row ID', 'Order Date'], axis=1)
```

3 OUR PROCESS

Data cleaning

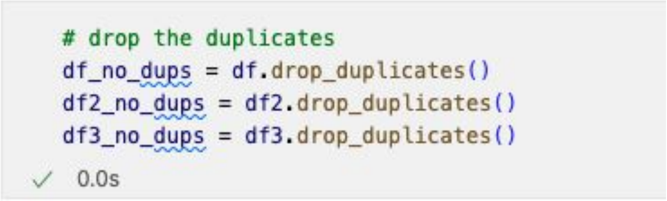


```
# check for duplicates
print('df=',df.duplicated().value_counts())
print('-' * 30)
print('df1=',df1.duplicated().value_counts())
print('-' * 30)
print('df2=',df2.duplicated().value_counts())
print('-' * 30)
print('df3=',df3.duplicated().value_counts())
```

```
df= False    9993
True          1
Name: count, dtype: int64
------------------------------
df1= False    333
Name: count, dtype: int64
------------------------------
df2= False    3003
True     2896
Name: count, dtype: int64
------------------------------
df3= False    204
True       86
Name: count, dtype: int64
```

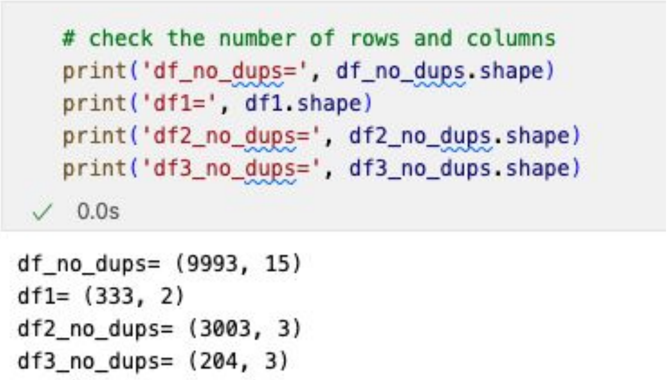
3 OUR PROCESS

Data cleaning



```
# drop the duplicates
df_no_dups = df.drop_duplicates()
df2_no_dups = df2.drop_duplicates()
df3_no_dups = df3.drop_duplicates()
```

✓ 0.0s



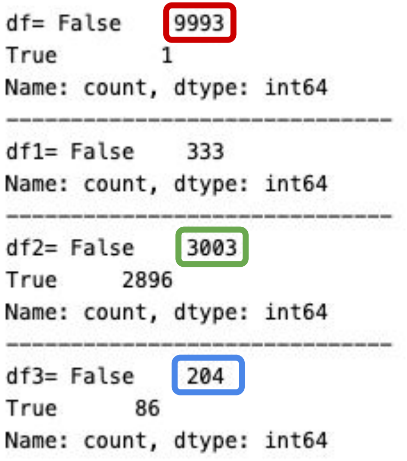
```
# check the number of rows and columns
print('df_no_dups=', df_no_dups.shape)
print('df1=', df1.shape)
print('df2_no_dups=', df2_no_dups.shape)
print('df3_no_dups=', df3_no_dups.shape)
```

✓ 0.0s

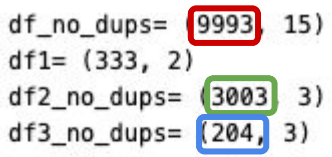
```
df_no_dups= (9993, 15)
df1= (333, 2)
df2_no_dups= (3003, 3)
df3_no_dups= (204, 3)
```

3 OUR PROCESS

Data cleaning



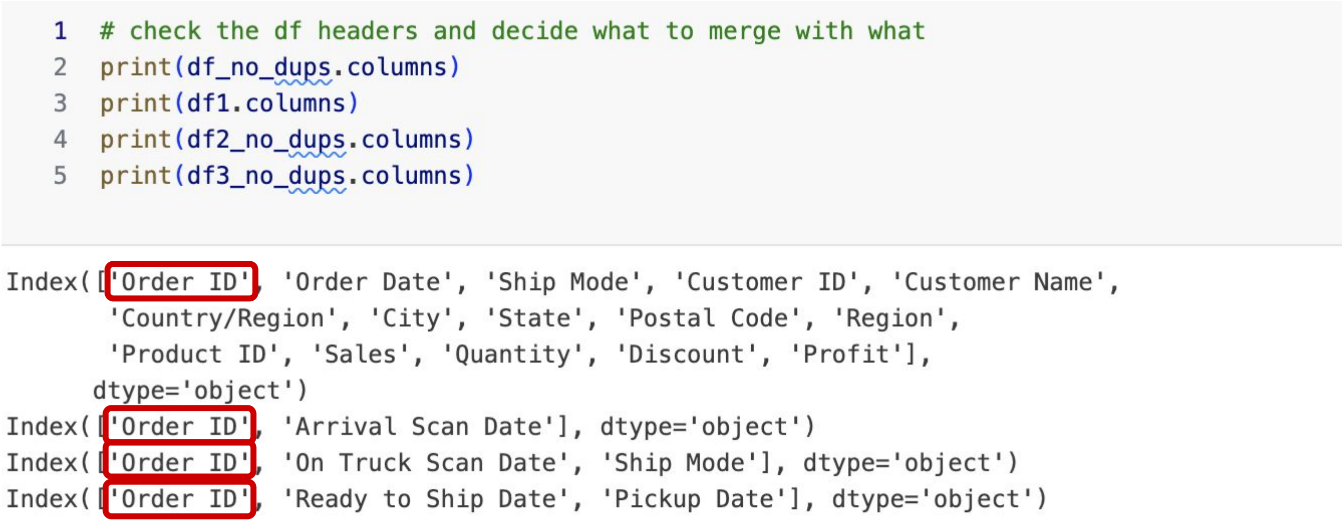
```
df= False     9993
True          1
Name: count, dtype: int64
------------------------------
df1= False    333
Name: count, dtype: int64
------------------------------
df2= False    3003
True      2896
Name: count, dtype: int64
------------------------------
df3= False    204
True       86
Name: count, dtype: int64
```



```
df_no_dups= (9993, 15)
df1= (333, 2)
df2_no_dups= (3003, 3)
df3_no_dups= (204, 3)
```

3 OUR PROCESS

Data merging



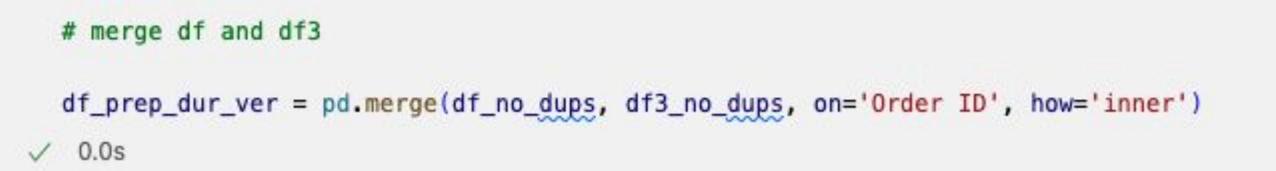
```python
# check the df headers and decide what to merge with what
print(df_no_dups.columns)
print(df1.columns)
print(df2_no_dups.columns)
print(df3_no_dups.columns)
```

```
Index(['Order ID', 'Order Date', 'Ship Mode', 'Customer ID', 'Customer Name',
       'Country/Region', 'City', 'State', 'Postal Code', 'Region',
       'Product ID', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
Index(['Order ID', 'Arrival Scan Date'], dtype='object')
Index(['Order ID', 'On Truck Scan Date', 'Ship Mode'], dtype='object')
Index(['Order ID', 'Ready to Ship Date', 'Pickup Date'], dtype='object')
```

Data cleaning

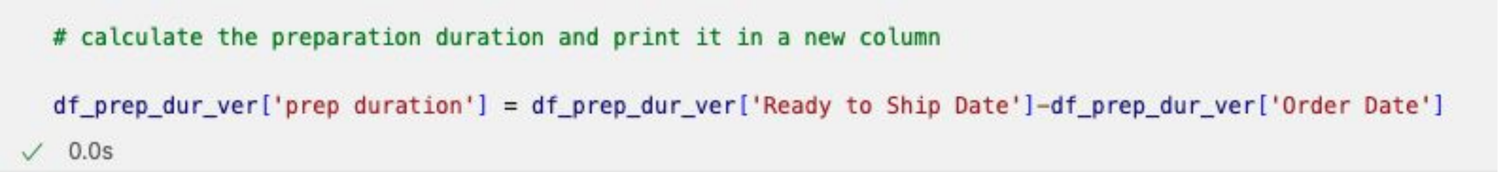
Order Fulfillment Process - add the dates we want to compare

1ST KPI - ORDER FULFILLMENT PROCESS



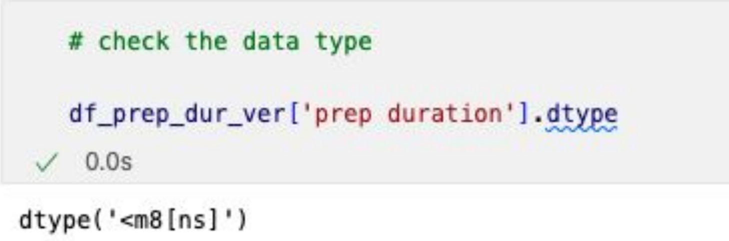
```
# merge df and df3

df_prep_dur_ver = pd.merge(df_no_dups, df3_no_dups, on='Order ID', how='inner')
```

✓ 0.0s



```
# calculate the preparation duration and print it in a new column

df_prep_dur_ver['prep duration'] = df_prep_dur_ver['Ready to Ship Date']-df_prep_dur_ver['Order Date']
```

✓ 0.0s



```
# check the data type

df_prep_dur_ver['prep duration'].dtype
```

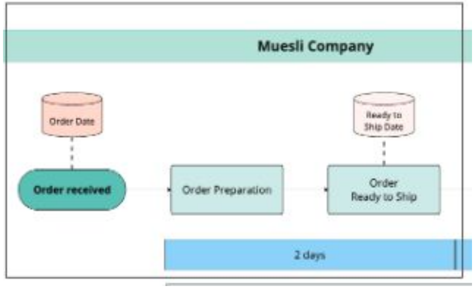
✓ 0.0s

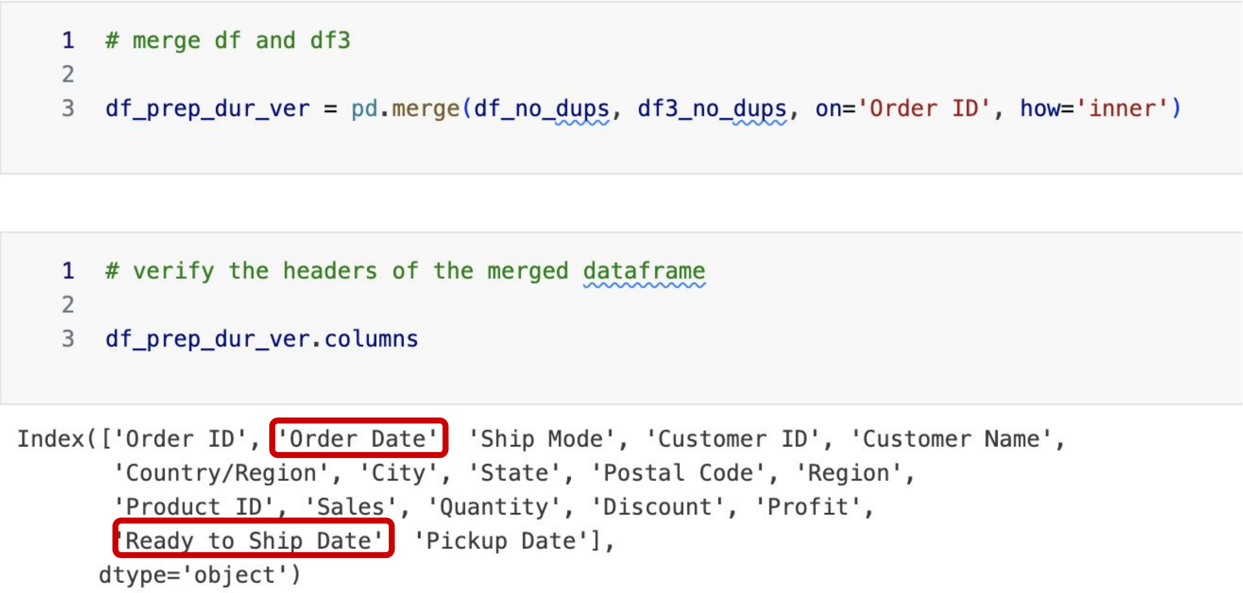
```
dtype('<m8[ns]')
```

3 OUR PROCESS

KPI 1 - Order Fulfillment Time

Order Fulfillment Process



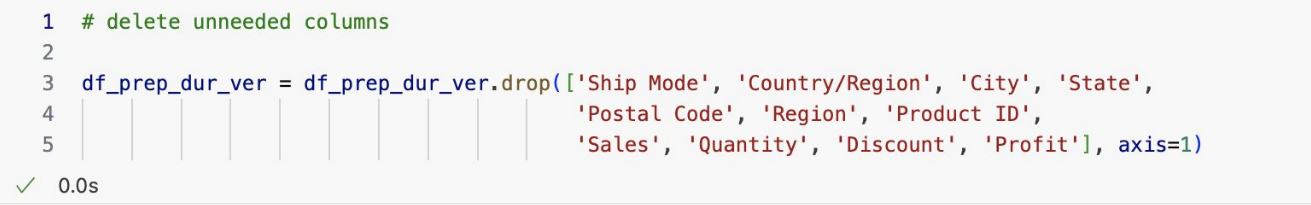


```
# merge df and df3

df_prep_dur_ver = pd.merge(df_no_dups, df3_no_dups, on='Order ID', how='inner')
```

```
# verify the headers of the merged dataframe

df_prep_dur_ver.columns
```

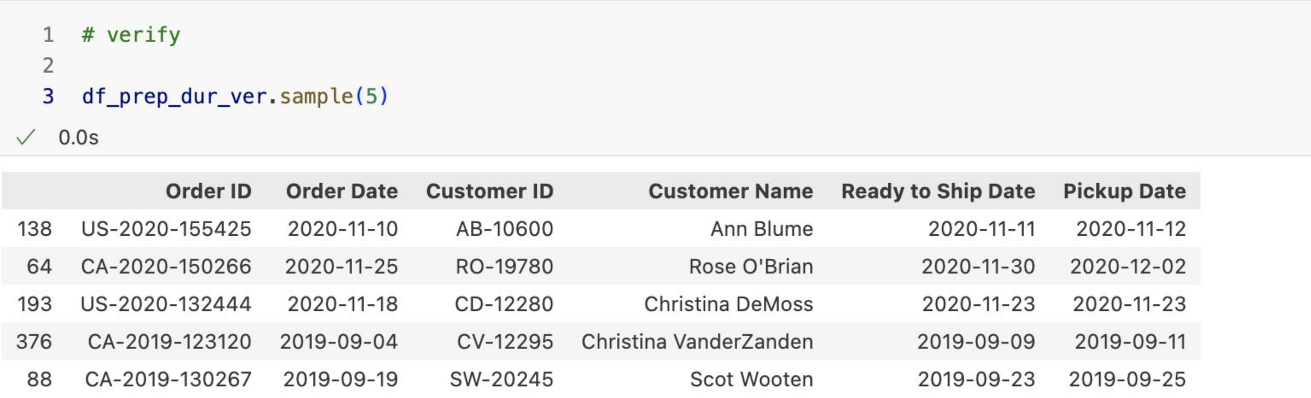
```
Index(['Order ID', 'Order Date', 'Ship Mode', 'Customer ID', 'Customer Name',
       'Country/Region', 'City', 'State', 'Postal Code', 'Region',
       'Product ID', 'Sales', 'Quantity', 'Discount', 'Profit',
       'Ready to Ship Date', 'Pickup Date'],
      dtype='object')
```

KPI 1 - Order Fulfillment Time



```python
# delete unneeded columns

df_prep_dur_ver = df_prep_dur_ver.drop(['Ship Mode', 'Country/Region', 'City', 'State',
                                        'Postal Code', 'Region', 'Product ID',
                                        'Sales', 'Quantity', 'Discount', 'Profit'], axis=1)
```

✓ 0.0s

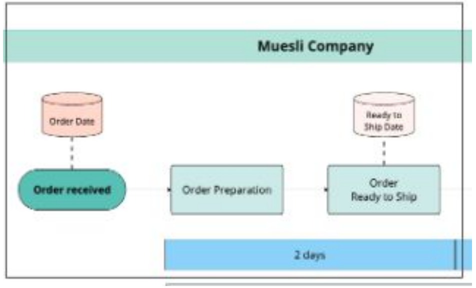


```python
# verify

df_prep_dur_ver.sample(5)
```

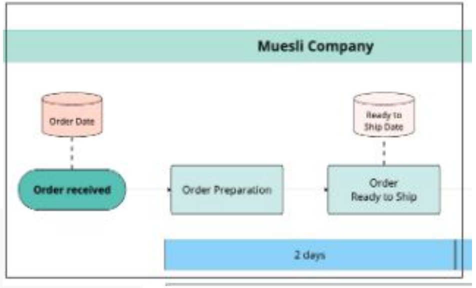
✓ 0.0s

	Order ID	Order Date	Customer ID	Customer Name	Ready to Ship Date	Pickup Date
138	US-2020-155425	2020-11-10	AB-10600	Ann Blume	2020-11-11	2020-11-12
64	CA-2020-150266	2020-11-25	RO-19780	Rose O'Brian	2020-11-30	2020-12-02
193	US-2020-132444	2020-11-18	CD-12280	Christina DeMoss	2020-11-23	2020-11-23
376	CA-2019-123120	2019-09-04	CV-12295	Christina VanderZanden	2019-09-09	2019-09-11
88	CA-2019-130267	2019-09-19	SW-20245	Scot Wooten	2019-09-23	2019-09-25

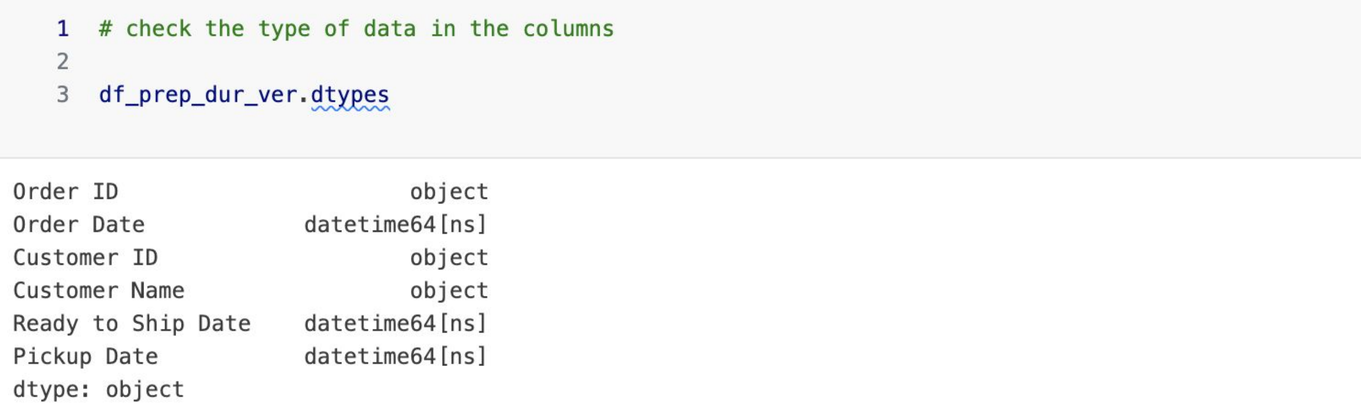
Order Fulfillment Process



KPI 1 - Order Fulfillment Time

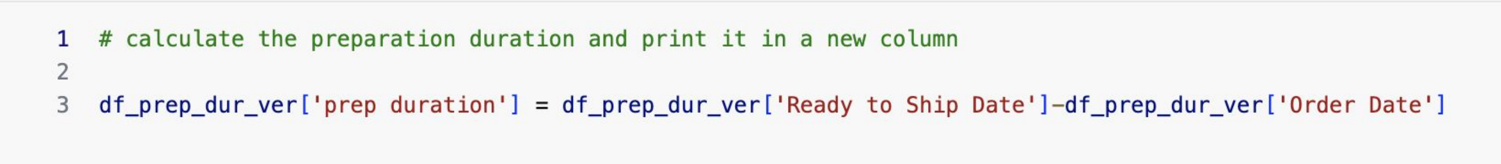


Order Fulfillment Process



```
# check the type of data in the columns

df_prep_dur_ver.dtypes
```

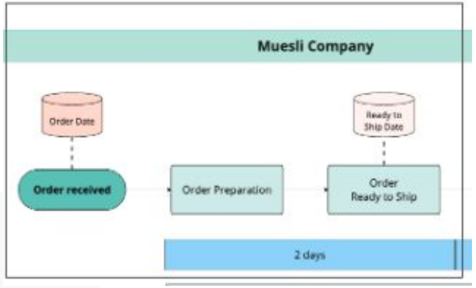
```
Order ID                         object
Order Date               datetime64[ns]
Customer ID                      object
Customer Name                    object
Ready to Ship Date       datetime64[ns]
Pickup Date              datetime64[ns]
dtype: object
```

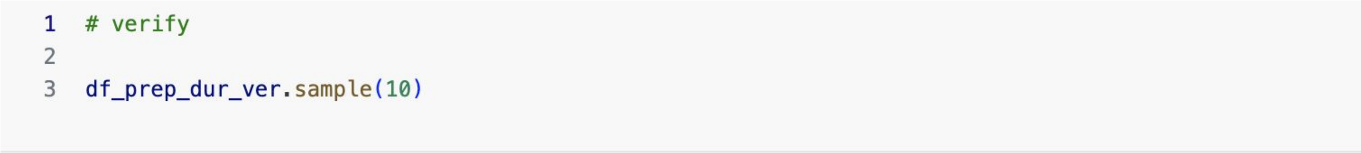


```
# calculate the preparation duration and print it in a new column

df_prep_dur_ver['prep duration'] = df_prep_dur_ver['Ready to Ship Date']-df_prep_dur_ver['Order Date']
```

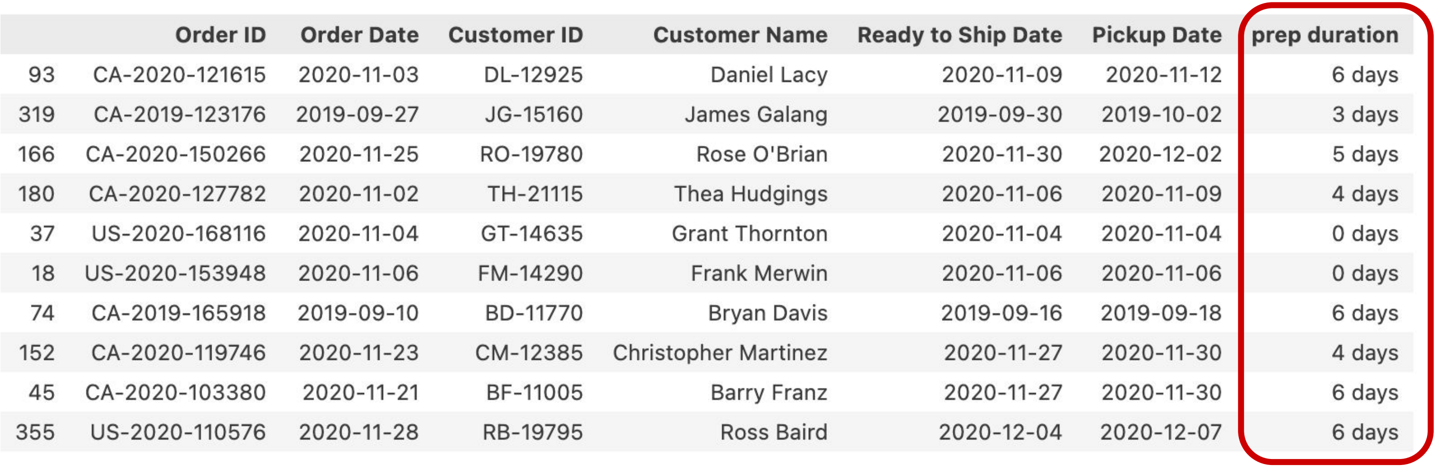
KPI 1 - Order Fulfillment Time

Order Fulfillment Process





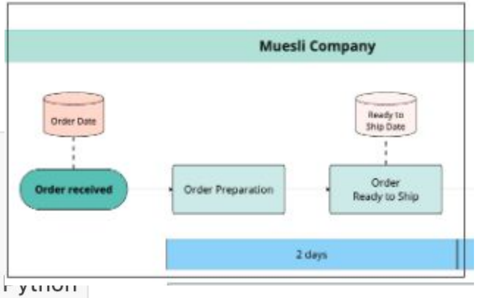
```
# verify

df_prep_dur_ver.sample(10)
```

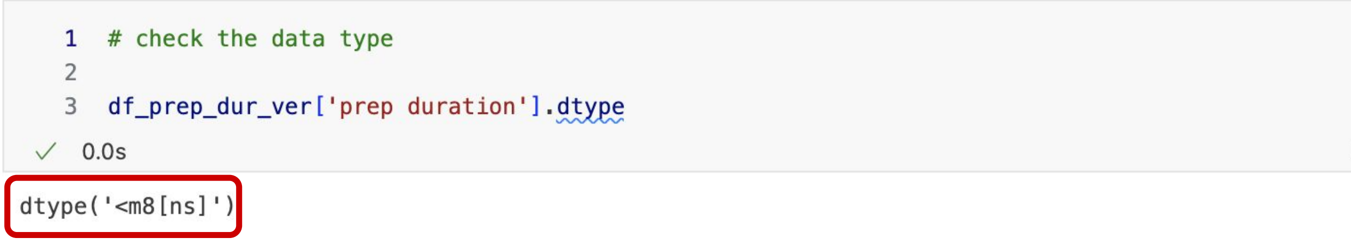


	Order ID	Order Date	Customer ID	Customer Name	Ready to Ship Date	Pickup Date	prep duration
93	CA-2020-121615	2020-11-03	DL-12925	Daniel Lacy	2020-11-09	2020-11-12	6 days
319	CA-2019-123176	2019-09-27	JG-15160	James Galang	2019-09-30	2019-10-02	3 days
166	CA-2020-150266	2020-11-25	RO-19780	Rose O'Brian	2020-11-30	2020-12-02	5 days
180	CA-2020-127782	2020-11-02	TH-21115	Thea Hudgings	2020-11-06	2020-11-09	4 days
37	US-2020-168116	2020-11-04	GT-14635	Grant Thornton	2020-11-04	2020-11-04	0 days
18	US-2020-153948	2020-11-06	FM-14290	Frank Merwin	2020-11-06	2020-11-06	0 days
74	CA-2019-165918	2019-09-10	BD-11770	Bryan Davis	2019-09-16	2019-09-18	6 days
152	CA-2020-119746	2020-11-23	CM-12385	Christopher Martinez	2020-11-27	2020-11-30	4 days
45	CA-2020-103380	2020-11-21	BF-11005	Barry Franz	2020-11-27	2020-11-30	6 days
355	US-2020-110576	2020-11-28	RB-19795	Ross Baird	2020-12-04	2020-12-07	6 days

KPI 1 - Order Fulfillment Time

Order Fulfillment Process





```
# check the data type

df_prep_dur_ver['prep duration'].dtype
```

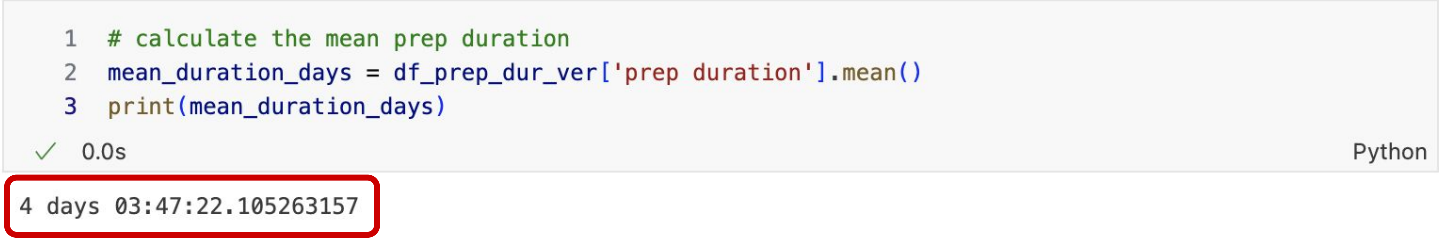
✓ 0.0s

```
dtype('<m8[ns]')
```

what?...

dtype('<m8[ns]') indicates that the data type of the object is a datetime64, representing date and time information with nanosecond precision.

ok, now I get it!



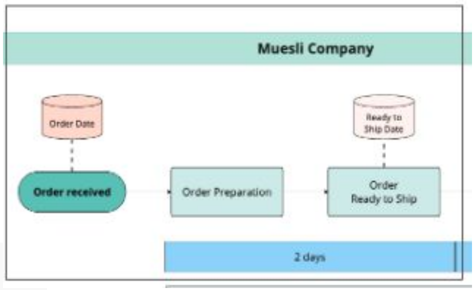
```
# calculate the mean prep duration
mean_duration_days = df_prep_dur_ver['prep duration'].mean()
print(mean_duration_days)
```

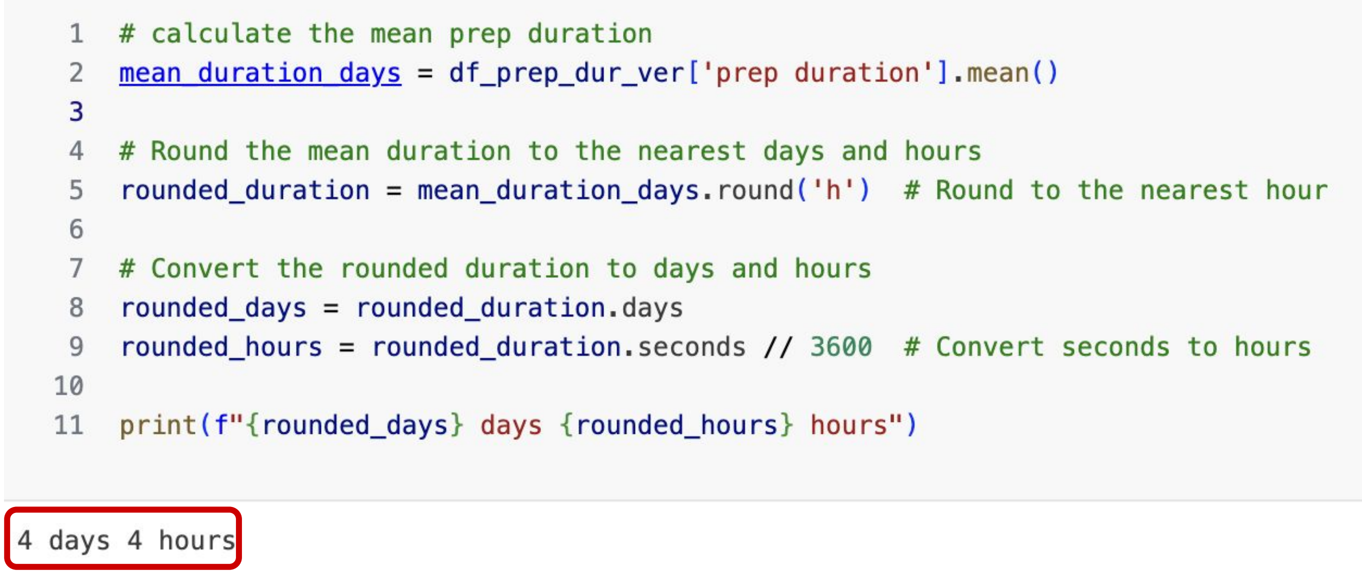
✓ 0.0s Python

```
4 days 03:47:22.105263157
```

KPI 1 - Order Fulfillment Time

Order Fulfillment Process

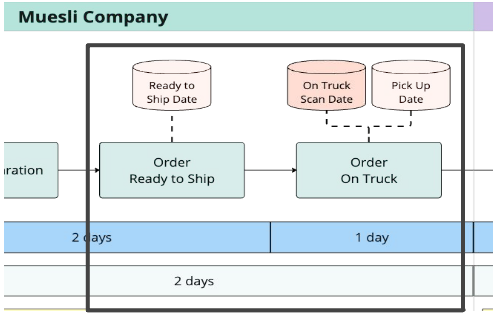


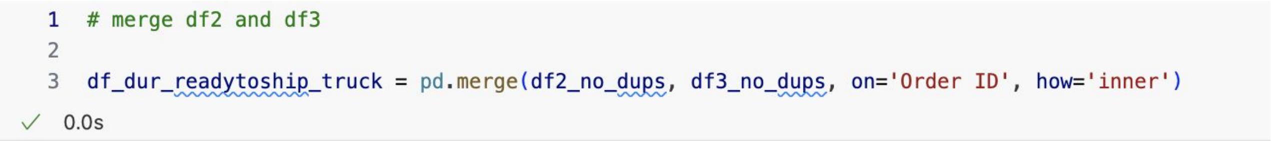


```python
# calculate the mean prep duration
mean_duration_days = df_prep_dur_ver['prep duration'].mean()

# Round the mean duration to the nearest days and hours
rounded_duration = mean_duration_days.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_days = rounded_duration.days
rounded_hours = rounded_duration.seconds // 3600  # Convert seconds to hours

print(f"{rounded_days} days {rounded_hours} hours")
```

```
4 days 4 hours
```

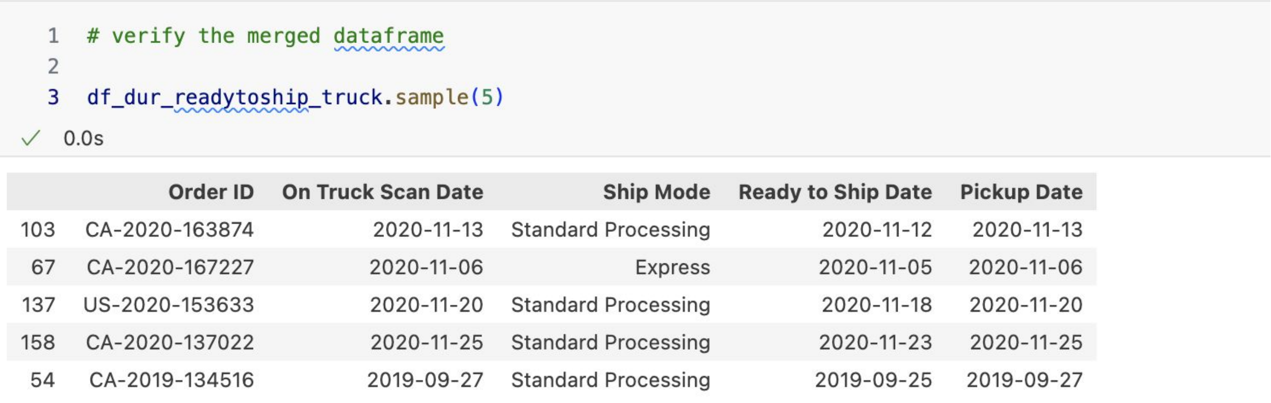
KPI 2 - Order Loading Time





```
# merge df2 and df3

df_dur_readytoship_truck = pd.merge(df2_no_dups, df3_no_dups, on='Order ID', how='inner')
```

✓ 0.0s

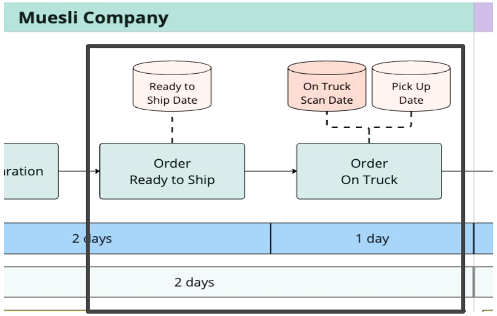


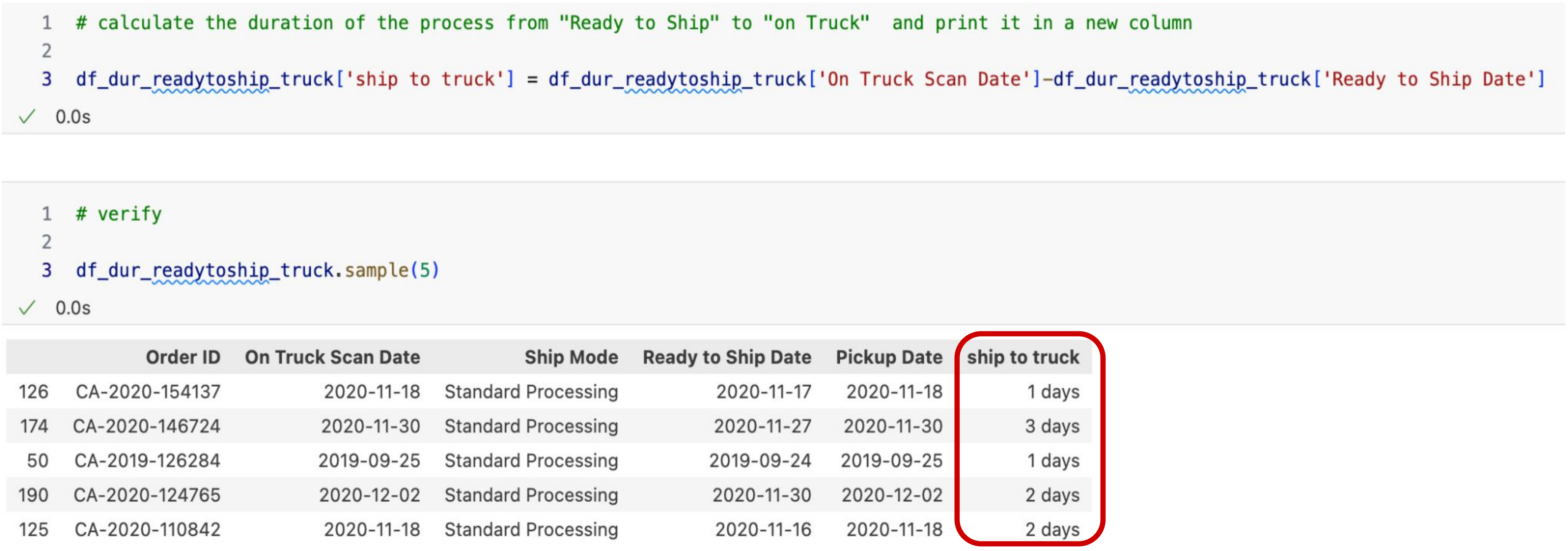
```
# verify the merged dataframe

df_dur_readytoship_truck.sample(5)
```

✓ 0.0s

	Order ID	On Truck Scan Date	Ship Mode	Ready to Ship Date	Pickup Date
103	CA-2020-163874	2020-11-13	Standard Processing	2020-11-12	2020-11-13
67	CA-2020-167227	2020-11-06	Express	2020-11-05	2020-11-06
137	US-2020-153633	2020-11-20	Standard Processing	2020-11-18	2020-11-20
158	CA-2020-137022	2020-11-25	Standard Processing	2020-11-23	2020-11-25
54	CA-2019-134516	2019-09-27	Standard Processing	2019-09-25	2019-09-27

KPI 2 - Order Loading Time





```python
# calculate the duration of the process from "Ready to Ship" to "on Truck"  and print it in a new column

df_dur_readytoship_truck['ship to truck'] = df_dur_readytoship_truck['On Truck Scan Date']-df_dur_readytoship_truck['Ready to Ship Date']
```

✓ 0.0s

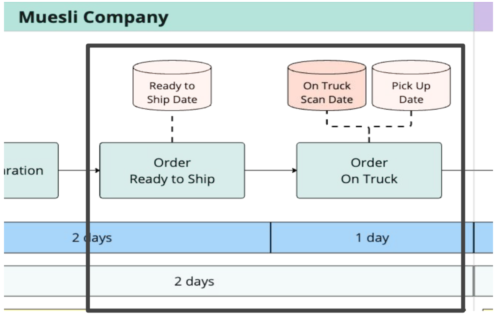
```python
# verify

df_dur_readytoship_truck.sample(5)
```

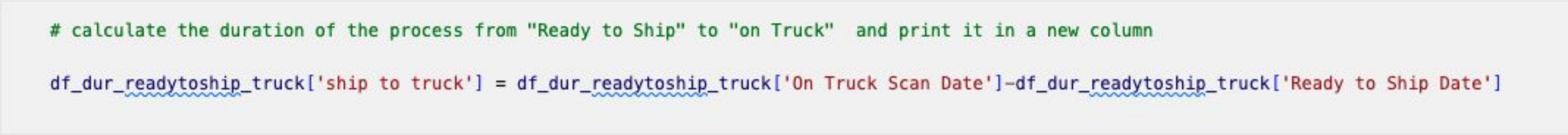
✓ 0.0s

	Order ID	On Truck Scan Date	Ship Mode	Ready to Ship Date	Pickup Date	ship to truck
126	CA-2020-154137	2020-11-18	Standard Processing	2020-11-17	2020-11-18	1 days
174	CA-2020-146724	2020-11-30	Standard Processing	2020-11-27	2020-11-30	3 days
50	CA-2019-126284	2019-09-25	Standard Processing	2019-09-24	2019-09-25	1 days
190	CA-2020-124765	2020-12-02	Standard Processing	2020-11-30	2020-12-02	2 days
125	CA-2020-110842	2020-11-18	Standard Processing	2020-11-16	2020-11-18	2 days

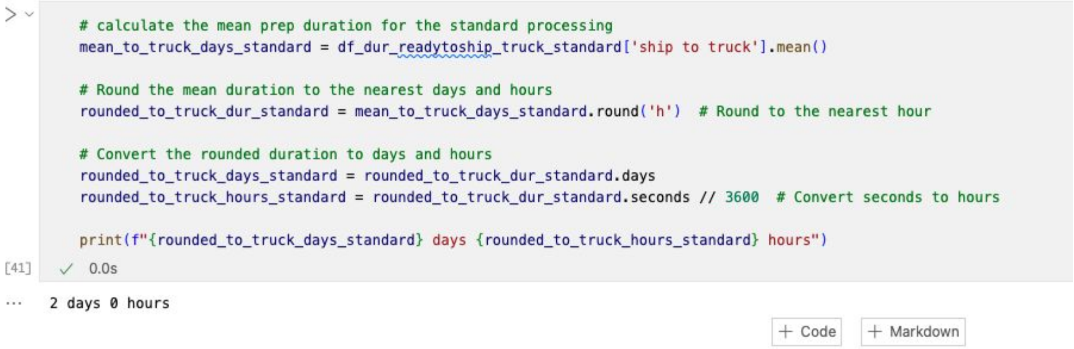
KPI 2

Order Loading Time





```
# calculate the duration of the process from "Ready to Ship" to "on Truck"  and print it in a new column

df_dur_readytoship_truck['ship to truck'] = df_dur_readytoship_truck['On Truck Scan Date']-df_dur_readytoship_truck['Ready to Ship Date']
```



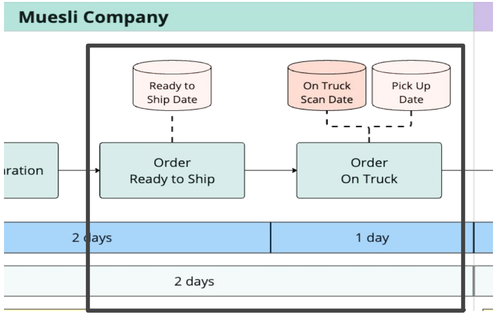
```
# calculate the mean prep duration for the standard processing
mean_to_truck_days_standard = df_dur_readytoship_truck_standard['ship to truck'].mean()

# Round the mean duration to the nearest days and hours
rounded_to_truck_dur_standard = mean_to_truck_days_standard.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_to_truck_days_standard = rounded_to_truck_dur_standard.days
rounded_to_truck_hours_standard = rounded_to_truck_dur_standard.seconds // 3600  # Convert seconds to hours

print(f"{rounded_to_truck_days_standard} days {rounded_to_truck_hours_standard} hours")
```

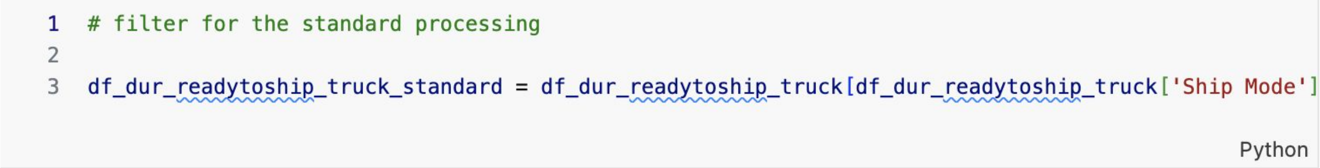
```
2 days 0 hours
```

...that's more than the claimed one day

KPI 2 - Order Loading Time

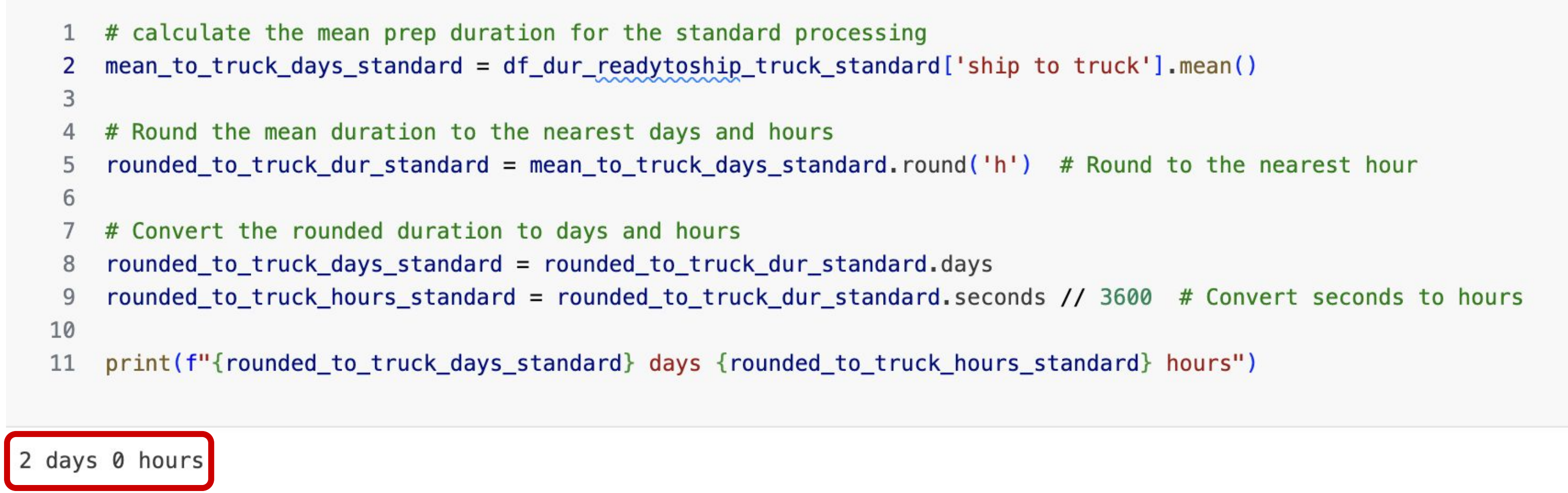
Standard Processing





```python
# filter for the standard processing

df_dur_readytoship_truck_standard = df_dur_readytoship_truck[df_dur_readytoship_truck['Ship Mode']
```

Python



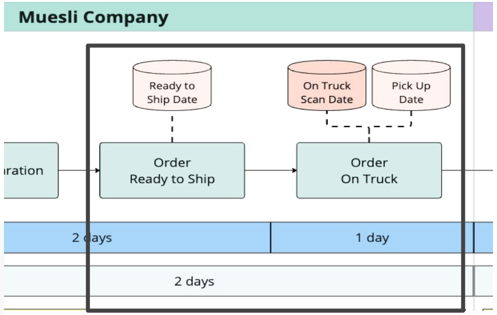
```python
# calculate the mean prep duration for the standard processing
mean_to_truck_days_standard = df_dur_readytoship_truck_standard['ship to truck'].mean()

# Round the mean duration to the nearest days and hours
rounded_to_truck_dur_standard = mean_to_truck_days_standard.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_to_truck_days_standard = rounded_to_truck_dur_standard.days
rounded_to_truck_hours_standard = rounded_to_truck_dur_standard.seconds // 3600  # Convert seconds to hours

print(f"{rounded_to_truck_days_standard} days {rounded_to_truck_hours_standard} hours")
```

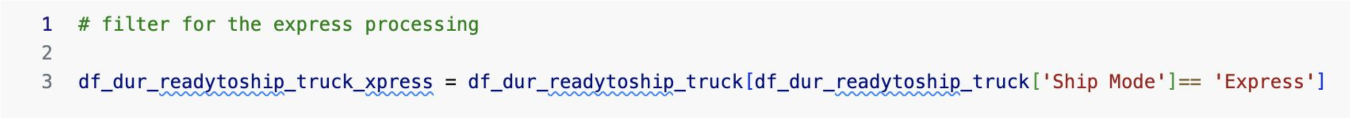
```
2 days 0 hours
```

...that's more than the claimed one day

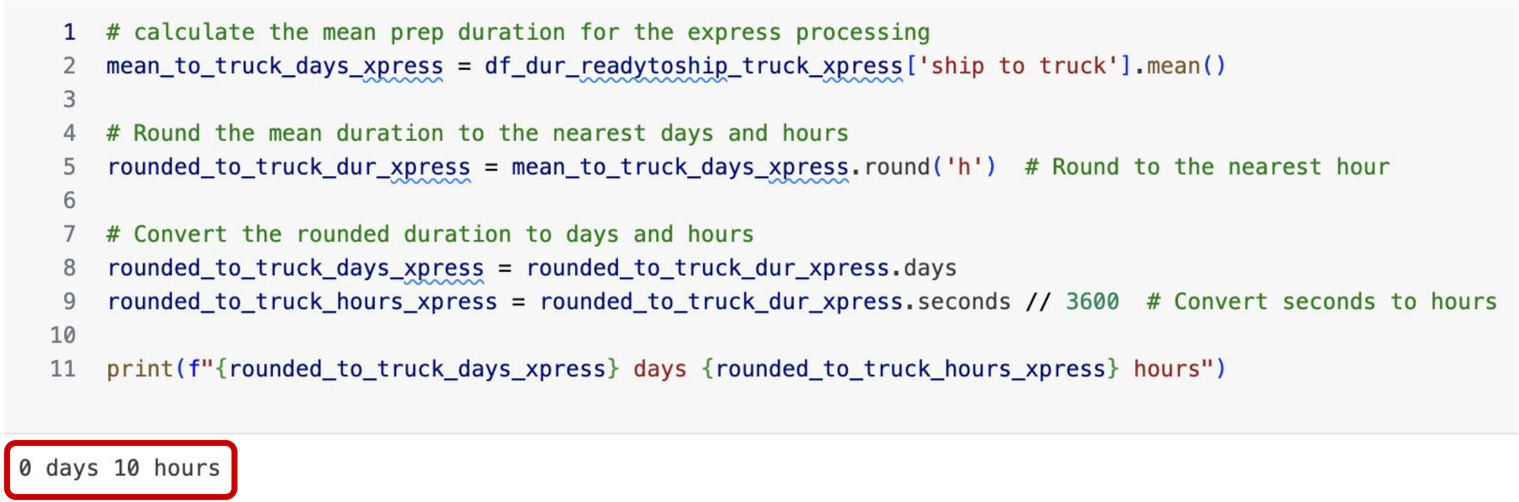
KPI 2 - Order Loading Time

Express Processing





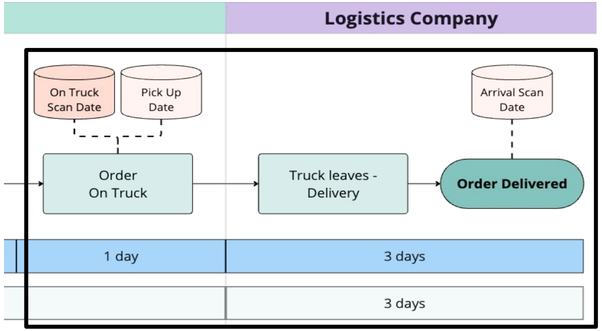
```python
# filter for the express processing

df_dur_readytoship_truck_xpress = df_dur_readytoship_truck[df_dur_readytoship_truck['Ship Mode']== 'Express']
```



```python
# calculate the mean prep duration for the express processing
mean_to_truck_days_xpress = df_dur_readytoship_truck_xpress['ship to truck'].mean()

# Round the mean duration to the nearest days and hours
rounded_to_truck_dur_xpress = mean_to_truck_days_xpress.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_to_truck_days_xpress = rounded_to_truck_dur_xpress.days
rounded_to_truck_hours_xpress = rounded_to_truck_dur_xpress.seconds // 3600  # Convert seconds to hours

print(f"{rounded_to_truck_days_xpress} days {rounded_to_truck_hours_xpress} hours")
```

```
0 days 10 hours
```

no comments



Logistics Company

On Truck Scan Date

Pick Up Date

Arrival Scan Date

Order On Truck

Truck leaves - Delivery

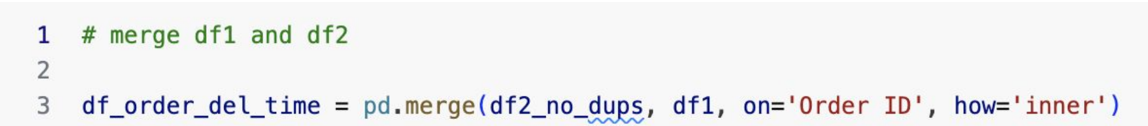
Order Delivered

1 day

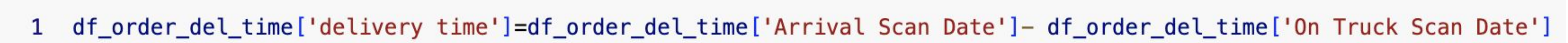
3 days

3 days

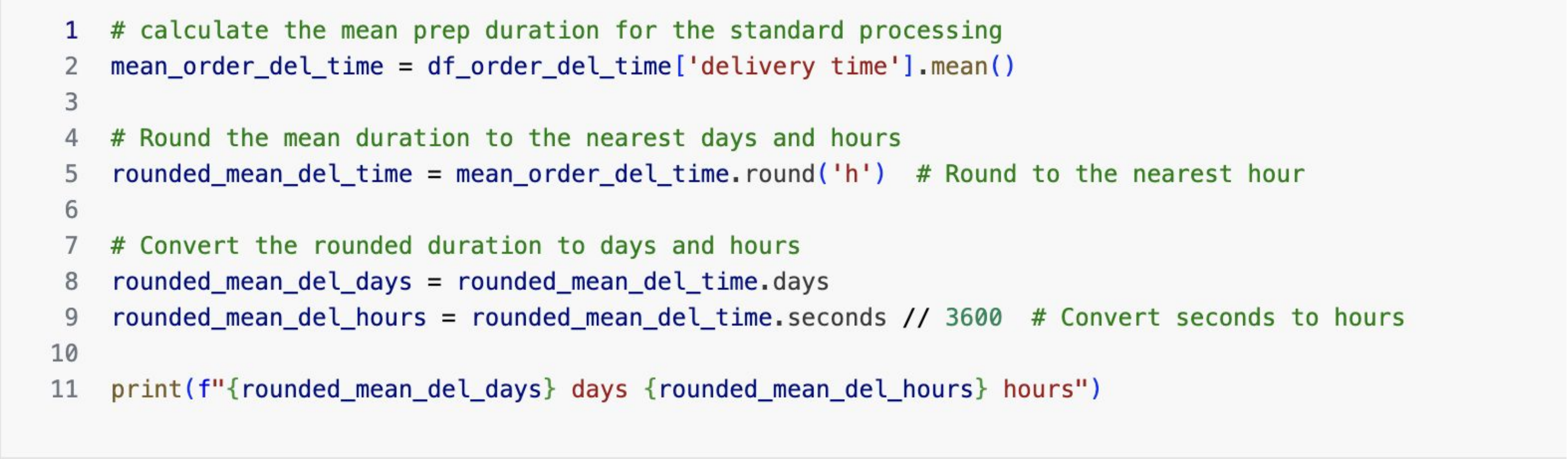
KPI 3 - Order Delivery Time



```
# merge df1 and df2

df_order_del_time = pd.merge(df2_no_dups, df1, on='Order ID', how='inner')
```



```
df_order_del_time['delivery time']=df_order_del_time['Arrival Scan Date']- df_order_del_time['On Truck Scan Date']
```

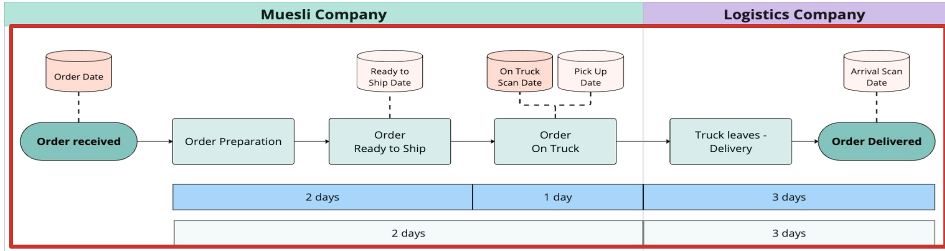


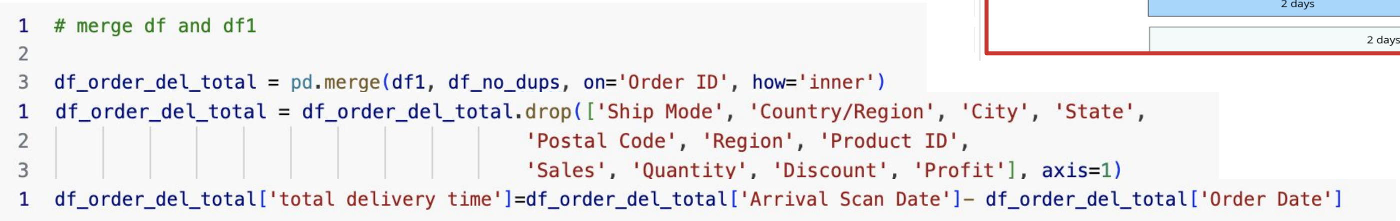
```
# calculate the mean prep duration for the standard processing
mean_order_del_time = df_order_del_time['delivery time'].mean()

# Round the mean duration to the nearest days and hours
rounded_mean_del_time = mean_order_del_time.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_mean_del_days = rounded_mean_del_time.days
rounded_mean_del_hours = rounded_mean_del_time.seconds // 3600  # Convert seconds to hours

print(f"{rounded_mean_del_days} days {rounded_mean_del_hours} hours")
```

```
4 days 14 hours
```

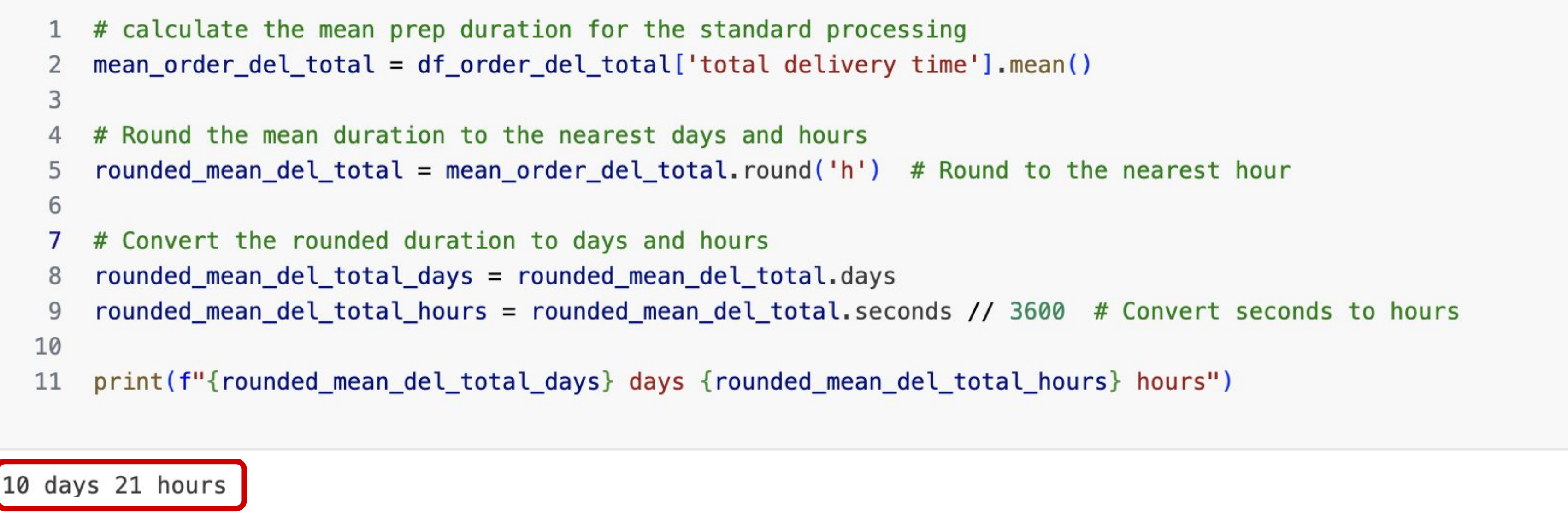
the shipping company claim is three days, oups!

KPI 4 - Order Processing Time





```
# merge df and df1

df_order_del_total = pd.merge(df1, df_no_dups, on='Order ID', how='inner')
df_order_del_total = df_order_del_total.drop(['Ship Mode', 'Country/Region', 'City', 'State',
                                        'Postal Code', 'Region', 'Product ID',
                                        'Sales', 'Quantity', 'Discount', 'Profit'], axis=1)
df_order_del_total['total delivery time']=df_order_del_total['Arrival Scan Date']- df_order_del_total['Order Date']
```



```
# calculate the mean prep duration for the standard processing
mean_order_del_total = df_order_del_total['total delivery time'].mean()

# Round the mean duration to the nearest days and hours
rounded_mean_del_total = mean_order_del_total.round('h')  # Round to the nearest hour

# Convert the rounded duration to days and hours
rounded_mean_del_total_days = rounded_mean_del_total.days
rounded_mean_del_total_hours = rounded_mean_del_total.seconds // 3600  # Convert seconds to hours

print(f"{rounded_mean_del_total_days} days {rounded_mean_del_total_hours} hours")
```

```
10 days 21 hours
```

AGENDA

1 ORDER PROCESS FLOW CHART

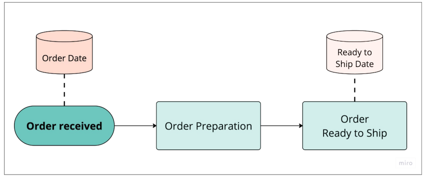
2 KPIs

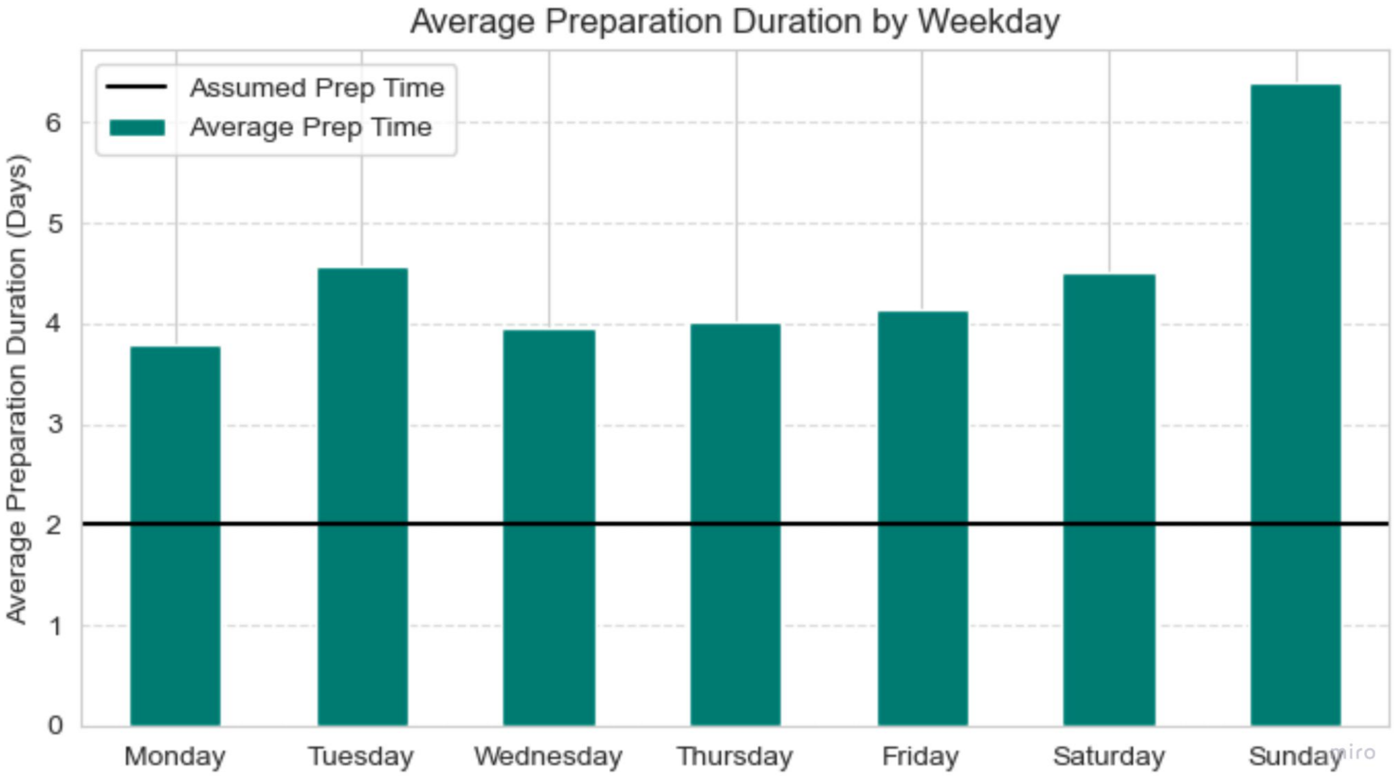
3 OUR PROCESS

4 OUR FINDINGS

5 CONCLUSIONS

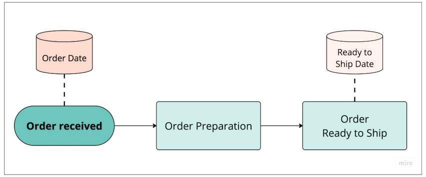
KPI 1 - Order Fulfillment Time

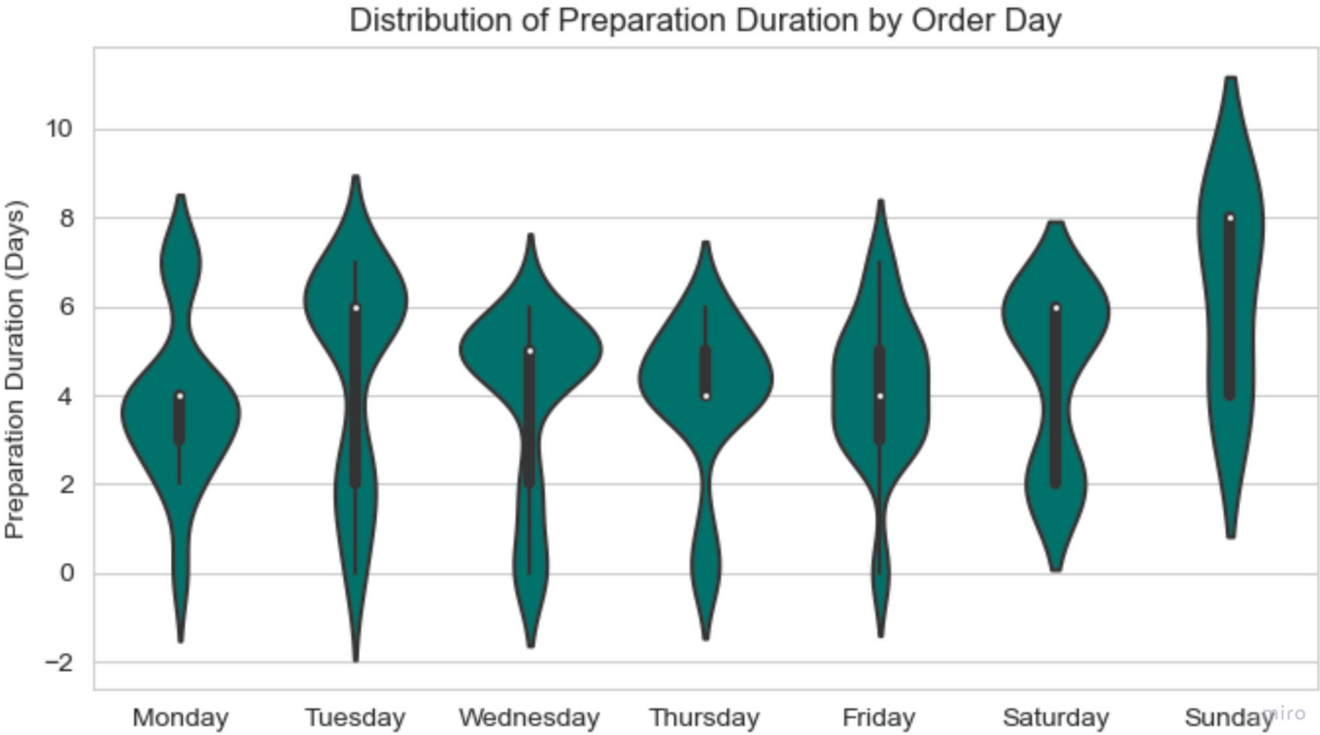




Average Preparation Time
4,5 Days

KPI 1 - Order Fulfillment Time

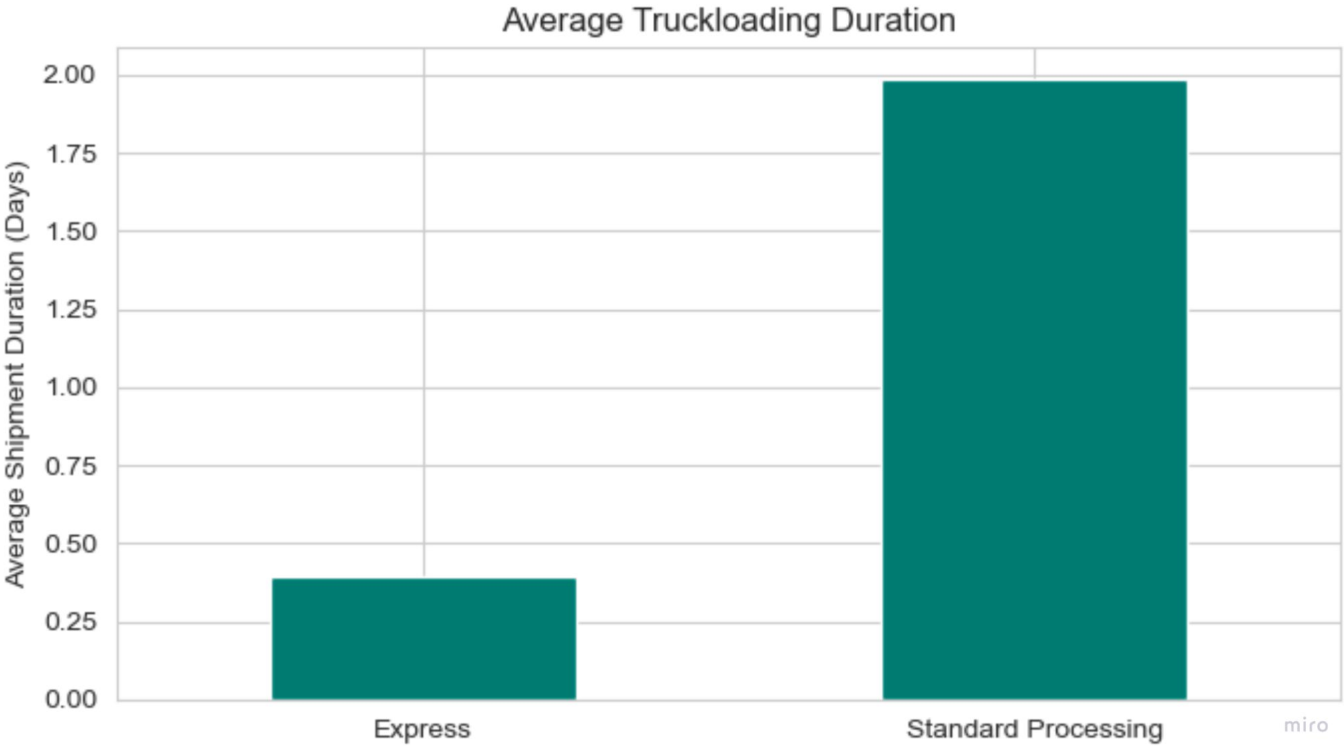


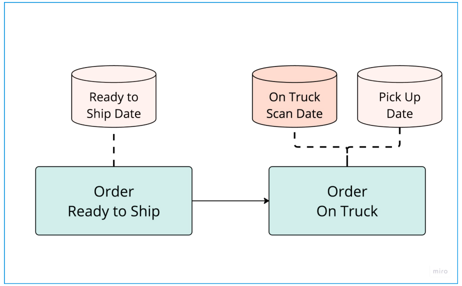


Preparation Time:

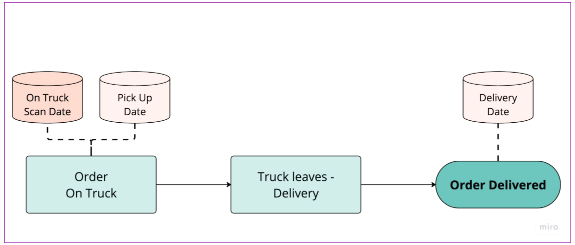
69%	late
20%	on time
11%	early

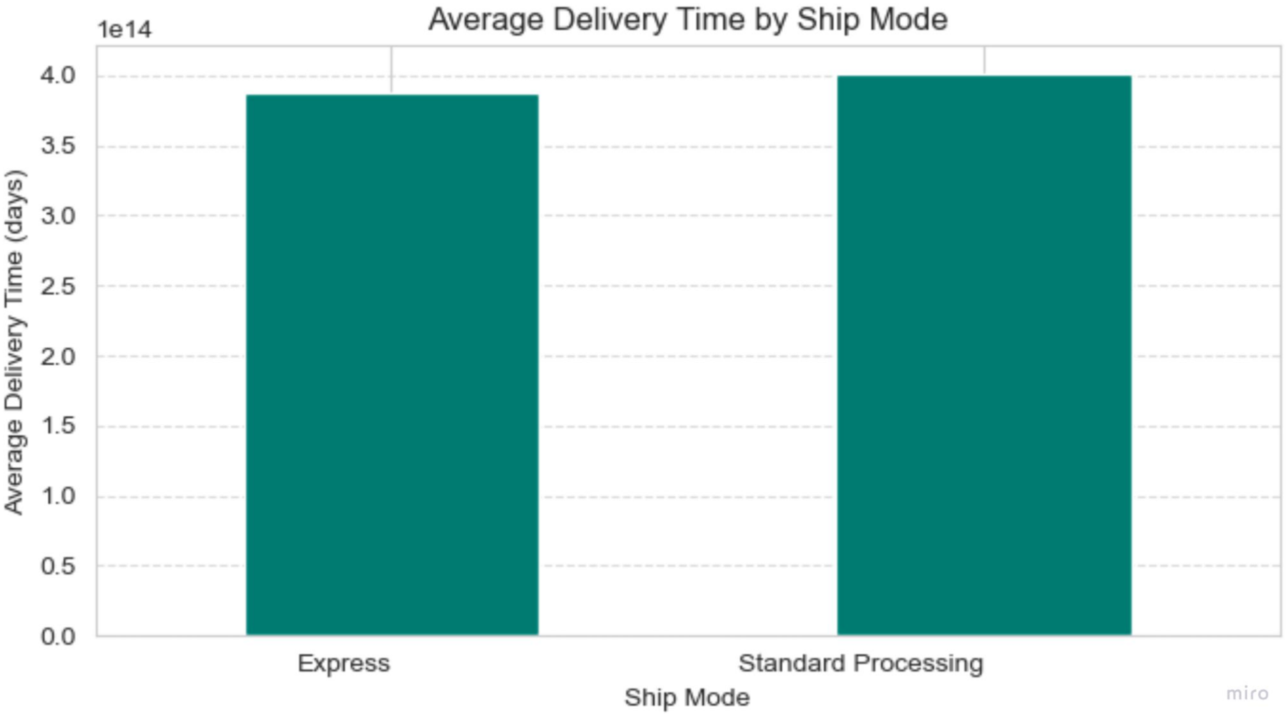
KPI 2 - Order Loading Time



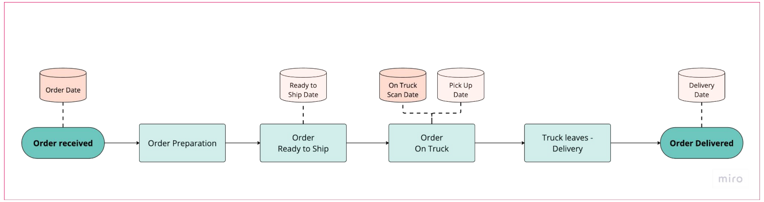


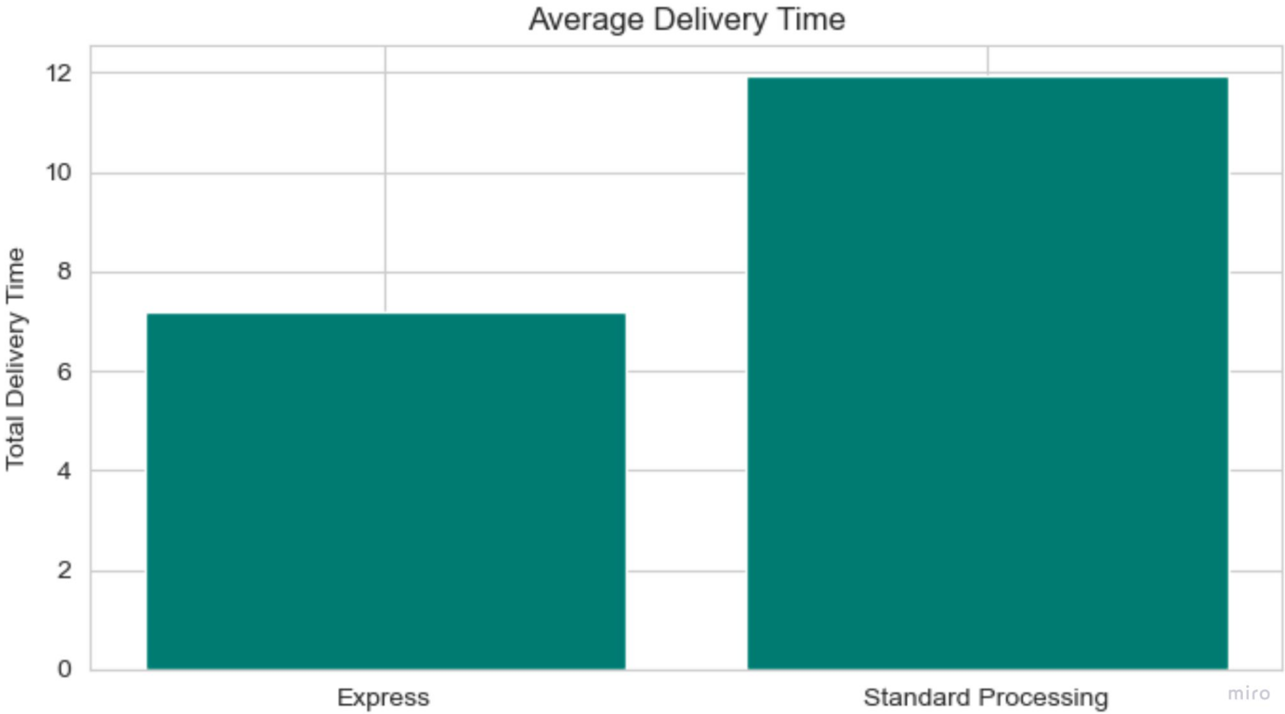
KPI 3 - Order Delivery Time





KPI 4 - Order Processing Time





AGENDA

1 ORDER PROCESS FLOW CHART

2 KPIs

3 OUR PROCESS

4 OUR FINDINGS

5 CONCLUSIONS

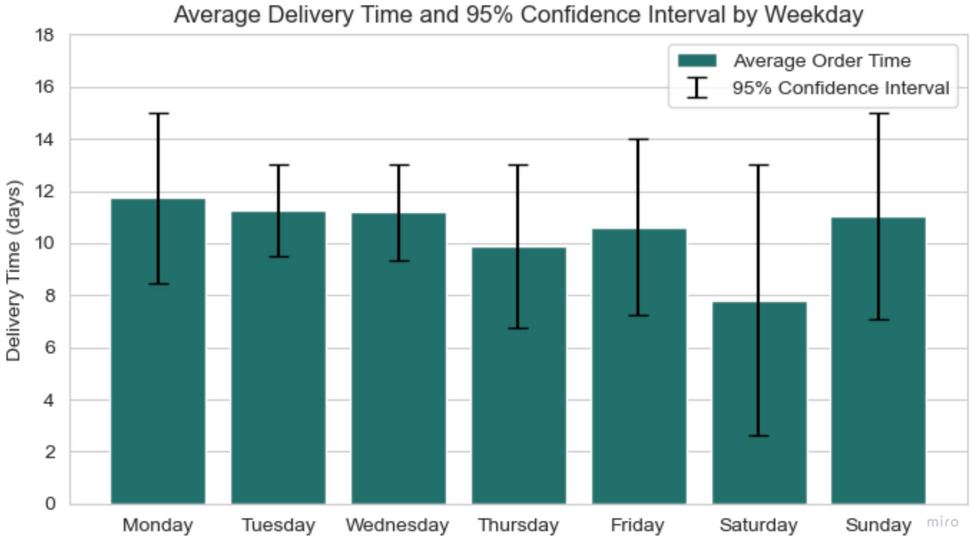
5 CONCLUSIONS

- **preparation of orders** in **warehouse** much longer than 2 days
  - only 20% of orders are prepared in 2 days, 69% are late
  - ↘ decrease preparation time in warehouse
- **express** in **order loading process** takes desired time
  - ↘ decrease loading time for standard to express time
- difference in **delivery time**: average 4 days instead of 3 days
  - ↺ no accurate data - get real data from delivery company

5 CONCLUSIONS

overall process:

- ↭ no consistency for preparation time for orders
    - ↘ same time period for every weekday



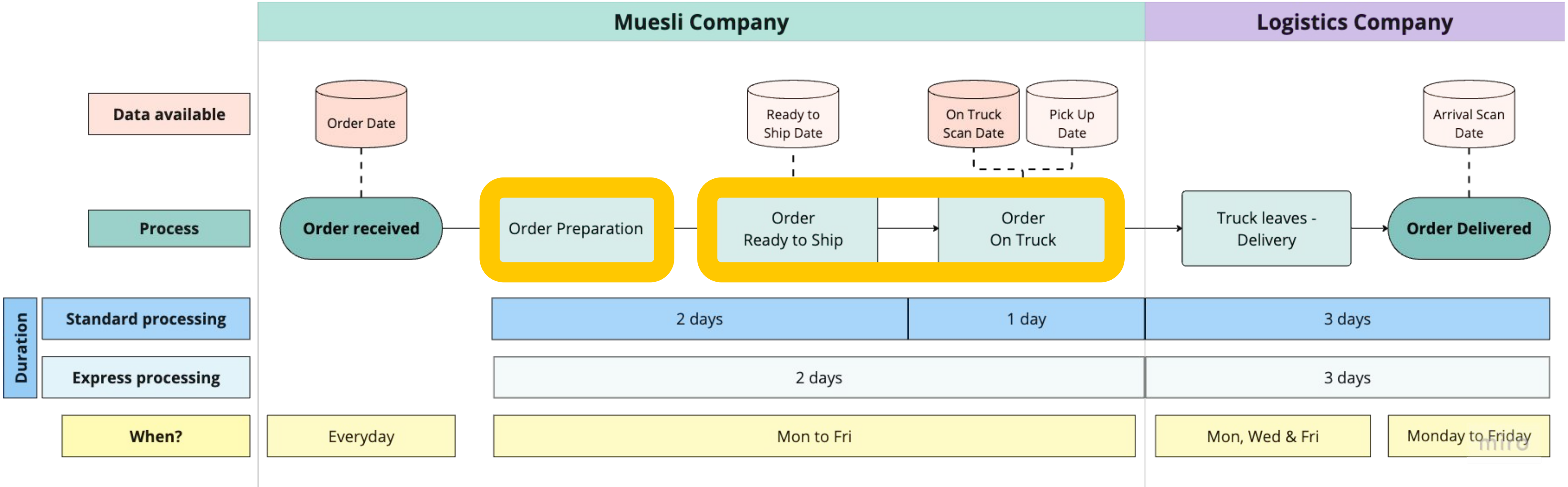
5 CONCLUSIONS

overall process:

- ↭ no consistency for preparation time for orders
  - ↘ same time period for every weekday
- in summary express delivery is faster than standard, though no difference in loading time (as assumed)
  - → apparently faster in warehouse

5 OVERALL CONCLUSIONS

- check efficiency of workflow (warehouse and loading)
- increase the amount of measurements in warehouse process



Muesli Company

Logistics Company

Data available

Order Date

Ready to Ship Date

On Truck Scan Date

Pick Up Date

Arrival Scan Date

Process

Order received

Order Preparation

Order Ready to Ship

Order On Truck

Truck leaves - Delivery

Order Delivered

Duration

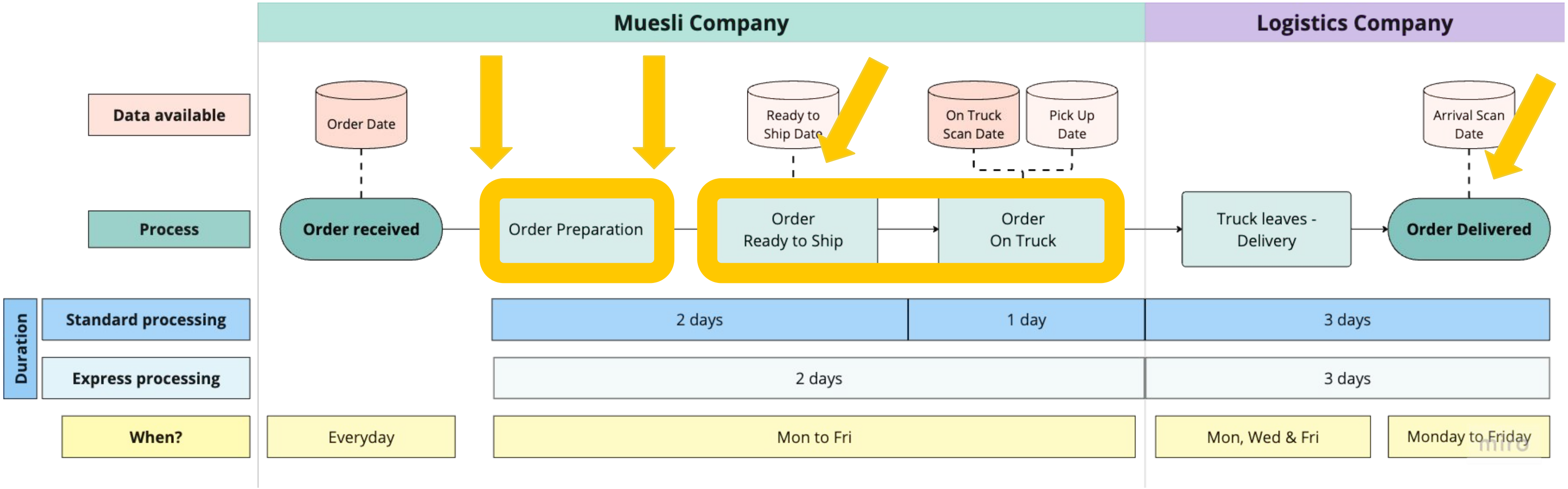
Standard processing: 2 days | 1 day | 3 days

Express processing: 2 days | 3 days

When?: Everyday | Mon to Fri | Mon, Wed & Fri | Monday to Friday

5 OVERALL CONCLUSIONS

- check efficiency of workflow (warehouse and loading)
- increase the amount of measurements in warehouse process



Muesli Company

Logistics Company

Data available: Order Date; Ready to Ship Date; On Truck Scan Date; Pick Up Date; Arrival Scan Date

Process: Order received → Order Preparation → Order Ready to Ship → Order On Truck → Truck leaves - Delivery → Order Delivered

Duration

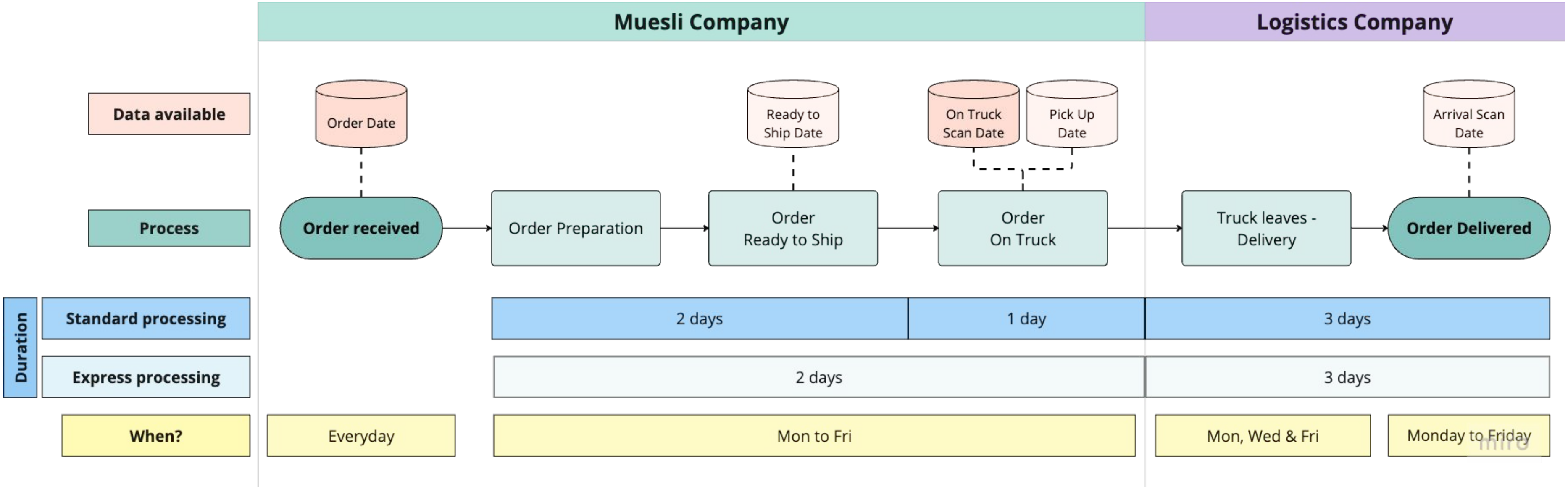
Standard processing: 2 days; 1 day; 3 days

Express processing: 2 days; 3 days

When?: Everyday; Mon to Fri; Mon, Wed & Fri; Monday to Friday

4 Overall Conclusions

- check efficiency of workflow (warehouse and loading)
- increase the amount of measurements in warehouse process
- increase workforce
- something with the delivery company



Muesli Company

Logistics Company

Data available

Order Date

Ready to Ship Date

On Truck Scan Date

Pick Up Date

Arrival Scan Date

Process

Order received

Order Preparation

Order Ready to Ship

Order On Truck

Truck leaves - Delivery

Order Delivered

Duration

Standard processing

2 days

1 day

3 days

Express processing

2 days

3 days

When?

Everyday

Mon to Fri

Mon, Wed & Fri

Monday to Friday

TIME FOR QUESTIONS