

# Final Assignment - Introduction to Data Warehousing



Estimated time needed: **90** minutes.

## About This SN Labs Cloud IDE

This Skills Network Labs Cloud IDE provides a hands-on environment for course and project related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform, that can be run on desktop or on the cloud. To complete this lab, we will be using the Cloud IDE based on Theia and PostgreSQL and MySQL database running in a Docker container. You will also need an instance of DB2 running in IBM Cloud.

## Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persistent. A new environment is created for you every time you connect to this lab. Any data you may have saved in an earlier session will get lost. To avoid losing your data, please plan to complete these labs in a single session.

## Scenario

You are a data engineer hired by a solid waste management company. The company collects and recycles solid waste across major cities in the country of Brazil. The company operates hundreds of trucks of different types to collect and transport solid waste. The company would like to create a data warehouse so that it can create reports like

- total waste collected per year per city
- total waste collected per month per city
- total waste collected per quarter per city
- total waste collected per year per trucktype
- total waste collected per trucktype per city
- total waste collected per trucktype per station per city

You will use your data warehousing skills to design and implement a data warehouse for the company.

## Objectives

In this assignment you will:

- Design a Data Warehouse
- Load data into Data Warehouse
- Write aggregation queries
- Create MQTs
- Create a Dashboard

## Note - Screenshots

Throughout this lab you will be prompted to take screenshots and save them on your own device. These screenshots will need to be uploaded for peer review in the next section of the course. You can use various free screengrabbing tools or your operating system's shortcut keys (Alt + PrintScreen in Windows, Command + Shift + 5 on Mac, Shift + Ctrl + Show windows on Chromebook) to capture the required screenshots. The screenshots can be either jpg or png.

## About the dataset

The dataset you would be using in this assignment is not a real life dataset. It was programmatically created for this assignment purpose.

## Prerequisites

You need access to a cloud instance of IBM DB2 to proceed with this assignment.

If you do not have an instance of IBM DB2 on cloud, follow the instructions in this [lab](#) to create one.

You need access to Cognos Analytics.

This [lab](#) will guide to get your access to Cognos Analytics, and also get you started with how to use it to analyze the data.

# Exercise 1 - Design a Data Warehouse

The solid waste management company has provied you the sample data they wish to collect.

Trip number	Waste Type	Waste Collected in tons	Collection Zone	City	Date
1	Dry	45.23	South	Sao Paulo	23-Jan-20
2	Wet	43.12	Central	Rio de Janeiro	24-Jan-20
3	Electronic	40.19	South	Sao Paulo	23-Jan-20
4	Plastic	34.87	West	Rio de Janeiro	24-Jan-20
5	Wet	45.34	West	Rio de Janeiro	23-Jan-20

You will start your project by designing a Star Schema warehouse by identifying the columns for the various dimension and fact tables in the schema.

## Task 1 - Design the dimension table MyDimDate

Write down the fields in the MyDimDate table in any text editor one field per line. The company is looking at a granularity of day. Which means they would like to have the ability to generate the report on yearly, monthly, daily, and weekday basis.

Here is a partial list of fields to serve as an example:

dateid  
month  
monthname  
...  
...

Take a screenshot of the fieldnames for the table MyDimDate.

Name the screenshot **1-MyDimDate.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 2 - Design the dimension table MyDimWaste

Write down the fields in the MyDimWaste table in any text editor one field per line.

Take a screenshot of the fieldnames for the table MyDimWaste.

Name the screenshot **2-MyDimWaste.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 3 - Design the dimension table MyDimZone

Write down the fields in the MyDimZone table in any text editor one field per line.

Take a screenshot of the fieldnames for the table MyDimZone.

Name the screenshot **3-MyDimZone.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 4 - Design the fact table MyFactTrips

Write down the fields in the MyFactTrips table in any text editor one field per line.

Take a screenshot of the fieldnames for the table MyFactTrips.

Name the screenshot **4-MyFactTrips.jpg**. (Images can be saved with either the .jpg or .png extension.)

# Exercise 2 - Create schema for Data Warehouse on DB2

In this exercise you will create the tables, you have designed in the previous exercise.

## Task 5 - Create the dimension table MyDimDate

Create the MyDimDate table.

Take a screenshot of the sql statement you used to create the table MyDimDate.

Name the screenshot **5-MyDimDate.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 6 - Create the dimension table MyDimWaste

Create the MyDimWaste table.

Take a screenshot of the sql statement you used to create the table MyDimWaste.

Name the screenshot **6-MyDimWaste.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 7 - Create the dimension table MyDimZone

Create the MyDimZone table.

Take a screenshot of the sql statement you used to create the table MyDimZone.

Name the screenshot **7-MyDimZone.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 8 - Create the fact table MyFactTrips

Create the MyFactTrips table.

Take a screenshot of the sql statement you used to create the table MyFactTrips.

Name the screenshot **8-MyFactTrips.jpg**. (Images can be saved with either the .jpg or .png extension.)

# Exercise 3 - Load data into the Data Warehouse

In this exercise you will load the data into the tables.

After the initial schema design, you were told that due to operational issues, data could not be collected in the format initially planned.

You will load the data provided by the company in csv format.

## Task 9 - Load data into the dimension table DimDate

Download the data from <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimDate.csv>

Load this data into DimDate table.

Take a screenshot of the first 5 rows in the table DimDate.

Name the screenshot **9-DimDate.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 10 - Load data into the dimension table DimTruck

Download the data from <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimTruck.csv>

Load this data into DimTruck table.

Take a screenshot of the first 5 rows in the table DimTruck.

Name the screenshot **10-DimTruck.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 11 - Load data into the dimension table DimStation

Download the data from <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimStation.csv>

Load this data into DimStation table.

Take a screenshot of the first 5 rows in the table DimStation.

Name the screenshot **11-DimStation.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 12 - Load data into the fact table FactTrips

Download the data from <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/FactTrips.csv>

Load this data into FactTrips table.

Take a screenshot of the first 5 rows in the table FactTrips.

Name the screenshot **12-FactTrips.jpg**. (Images can be saved with either the .jpg or .png extension.)

# Exercise 4 - Write aggregation queries and create MQTs

In this exercise you will query the data you have loaded in the previous exercise.

## Task 13 - Create a grouping sets query

Create a grouping sets query using the columns stationid, trucktype, total waste collected.

Take a screenshot of the sql and the output rows.

Name the screenshot **13-groupingsets.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 14 - Create a rollup query

Create a rollup query using the columns year, city, stationid, and total waste collected.

Take a screenshot of the sql and the output rows.

Name the screenshot **14-rollup.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 15 - Create a cube query

Create a cube query using the columns year, city, stationid, and average waste collected.

Take a screenshot of the sql and the output rows.

Name the screenshot **15-cube.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 16 - Create an MQT

Create an MQT named max\_waste\_stats using the columns city, stationid, trucktype, and max waste collected.

Take a screenshot of the sql.

Name the screenshot **16-mqt.jpg**. (Images can be saved with either the .jpg or .png extension.)

# Exercise 5 - Create a dashboard using Cognos Analytics

Download the data from <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DataForCognos.csv>

Use the DataForCognos.csv file to generate the following charts.

## Task 17 - Create a pie chart in the dashboard

Create a pie chart that shows the waste collected by truck type.

Take a screenshot of the pie chart.

Name the screenshot **17-pie.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 18 - Create a bar chart in the dashboard

Create a bar chart that shows the waste collected station wise.

Take a screenshot of the bar chart.

Name the screenshot **18-bar.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 19 - Create a line chart in the dashboard

Create a line chart that shows the waste collected by month wise.

Take a screenshot of the line chart.

Name the screenshot **19-line.jpg**. (Images can be saved with either the .jpg or .png extension.)

## Task 20 - Create a pie chart in the dashboard

Create a pie chart that shows the waste collected by city.

Take a screenshot of the pie chart.

Name the screenshot **20-pie.jpg**. (Images can be saved with either the .jpg or .png extension.)

End of the assignment.

## Authors

Ramesh Sannareddy

## Other Contributors

Rav Ahuja

## Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-09-29	0.1	Ramesh Sannareddy	Created initial version

2021-10-6	0.2	Steve Hord	Copy edits
-----------	-----	------------	------------

Copyright (c) 2021 IBM Corporation. All rights reserved.

## Task 1 - Design the dimension table MyDimDate

MyDimDate table

date\_id (PK)  
day  
month  
month\_name  
week\_day  
week\_number  
year

## Task 2 - Design the dimension table MyDimWaste

MyDimWaste table

waste\_id (PK)  
waste\_type

## Task 3 - Design the dimension table MyDimZone

MyDimZone Table

collection\_zone\_id (PK)  
collection\_zone  
city


## Task 4 - Design the fact table MyFactTrips

MyFactsTrip

trip\_id (PK)  
date\_id (FK)  
waste\_id (FK)  
collection\_zone\_id (FK)  
trip\_number

## Task 5 - Create the dimension table MyDimDate

```
CREATE TABLE MyDimDate (  
    date_id integer NOT NULL primary key,  
    day integer NOT NULL,  
    week_day varchar (10) NOT NULL,  
    week_number integer NOT NULL,  
    month integer NOT NULL,  
    month_name varchar (10) NOT NULL,  
    year integer NOT NULL  
);
```

^  CREATE TABLE MyDimDate ... Tempo de Execução: 0.222 s

Status: **Sucesso** | Linhas afetadas: 0

## Task 6 - Create the dimension table MyDimWaste'

```
CREATE TABLE MyDimWaste (  
    waste_id integer NOT NULL primary key,  
    waste_type varchar (10) NOT NULL  
);
```

## Task 7 - Create the dimension table MyDimZone

```
CREATE TABLE MyDimZone (  
    collection_zone_id integer NOT NULL primary k  
    collection_zone varchar (10) NOT NULL,  
    city varchar (10) NOT NULL  
);
```

## Task 8 - Create the fact table MyFactTrips

```
CREATE TABLE MyFactsTrip (  
    trip_id integer NOT NULL primary key,  
    date_id integer NOT NULL,  
    waste_id integer NOT NULL,  
    collection_zone_id integer NOT NULL,  
    trip_number integer NOT NULL  
);
```

## Task 9 - Load data into the dimension table DimDate

RWM48321.DIMDATE										Voltar
										Exportar para CSV
DATEID	DATE	YEAR	QUARTER	QUARTERNAME	MONTH	MONTHNAME	DAY	WEEKDAY	WEEKDAYNAME	
1	2019-03-09	2019	1	Q1	3	March	9	7	Sunday	
2	2019-03-10	2019	1	Q1	3	March	10	1	Monday	
3	2019-03-11	2019	1	Q1	3	March	11	2	Tuesday	
4	2019-03-12	2019	1	Q1	3	March	12	3	Wednesday	
5	2019-03-13	2019	1	Q1	3	March	13	4	Thursday	

## Task 10 - Load data into the dimension table DimTruck

RWM48321.DIMTRUCK	
TRUCKID	TRUCKTYPE
115	Volvo
120	Scania
121	Volvo
122	Scania
125	Volvo



## RWM48321.DIMSTATION

Volar

[Exportar para CSV](#)

	STATIONID INTEGER	CITY VARCHAR(32)
1	19	Sao Paulo
2	21	Sao Paulo
3	31	Rio de Janeiro
4	32	Rio de Janeiro
5	40	Brasilia

## RWM48321.FACTTRIPS

Volar

[Exportar para CSV](#)

	TRIPID INTEGER	DATED INTEGER	STATIONID INTEGER	TRUCKID INTEGER	WASTECOLLECTED REAL
1	23475	1	71	133	33.36
2	23476	1	46	162	34.88
3	23477	1	40	134	34.69
4	23478	1	43	148	30.01
5	23479	1	46	169	37.47

## Task 13 - Create a grouping sets query

\*Untitled - 1
\*Untitled - 2

```

1 select stationid, trucktype, sum(wastecollected) as totalwastecollected
2 from facttrips
3 left join dimtruck
4 on facttrips.truckid = dimtruck.truckid
5 group by grouping sets(stationid,trucktype)
6 order by stationid, trucktype;
7
8
    
```

Resultado: - Mar 23, 2022 8:11:42 PM

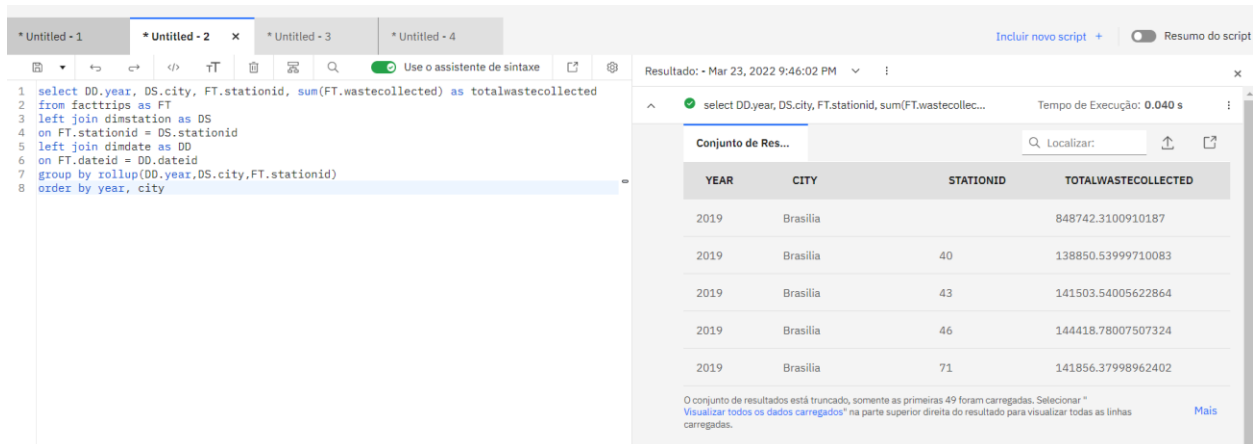
select stationid, trucktype, sum(wastecollected) as total... Tempo de Execução: 0.040 s

**Conjunto de Res...**

STATIONID	TRUCKTYPE	TOTALWASTECOLLECTED
19		332816.77986335754
21		333142.01016044617
31		167061.68010139465
32		167764.84003257751
40		163671.70993614197

O conjunto de resultados está truncado, somente as primeiras 21 foram carregadas. Selecionar "Visualizar todos os dados carregados" na parte superior direita do resultado para visualizar todas as linhas carregadas.

## Task 14 - Create a rollup query



The screenshot shows a SQL IDE with a query editor on the left and a results pane on the right. The query is a rollup query that calculates the total waste collected for each year, city, and station.

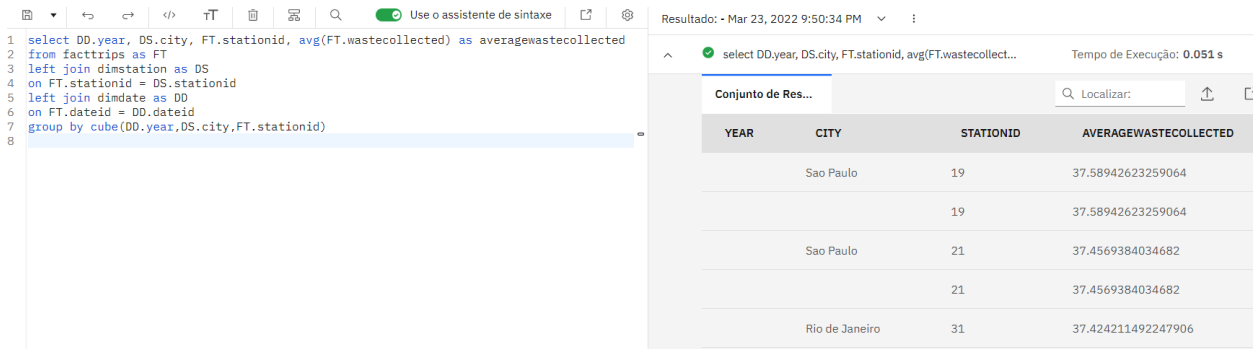
```
1 select DD.year, DS.city, FT.stationid, sum(FT.wastecollected) as totalwastecollected
2 from facttrips as FT
3 left join dimstation as DS
4 on FT.stationid = DS.stationid
5 left join dimdate as DD
6 on FT.dateid = DD.dateid
7 group by rollup(DD.year, DS.city, FT.stationid)
8 order by year, city
```

The results pane shows the following data:

YEAR	CITY	STATIONID	TOTALWASTECOLLECTED
2019	Brasilia		848742.3100910187
2019	Brasilia	40	138850.53999710083
2019	Brasilia	43	141503.54005622864
2019	Brasilia	46	144418.78007507324
2019	Brasilia	71	141856.37998962402

O conjunto de resultados está truncado, somente as primeiras 49 foram carregadas. Selecionar "Visualizar todos os dados carregados" na parte superior direita do resultado para visualizar todas as linhas carregadas. [Mais](#)

## Task 15 - Create a cube query



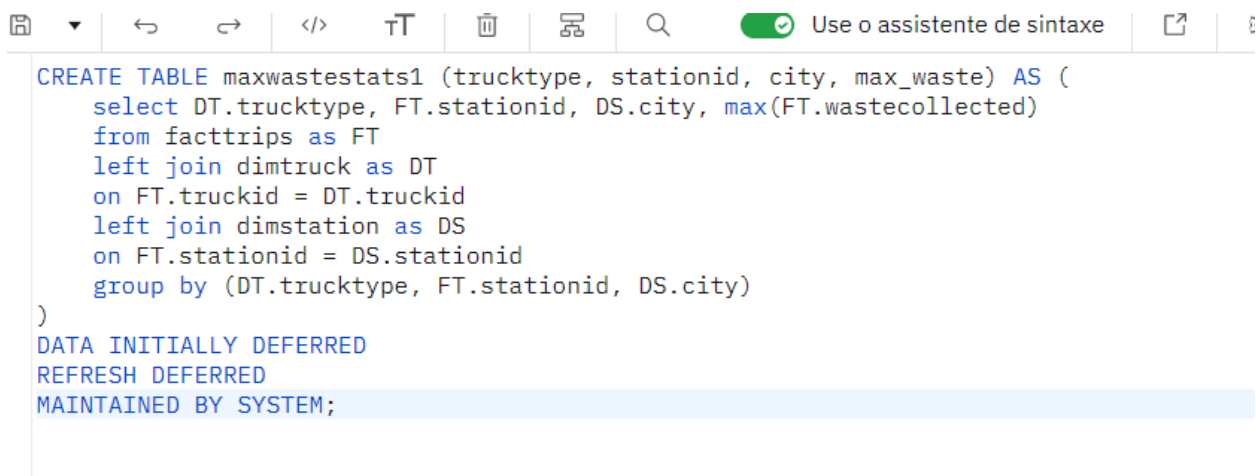
The screenshot shows a SQL IDE with a query editor on the left and a results pane on the right. The query is a cube query that calculates the average waste collected for each year, city, and station.

```
1 select DD.year, DS.city, FT.stationid, avg(FT.wastecollected) as averagewastecollected
2 from facttrips as FT
3 left join dimstation as DS
4 on FT.stationid = DS.stationid
5 left join dimdate as DD
6 on FT.dateid = DD.dateid
7 group by cube(DD.year, DS.city, FT.stationid)
8
```

The results pane shows the following data:

YEAR	CITY	STATIONID	AVERAGEWASTECOLLECTED
	Sao Paulo	19	37.58942623259064
		19	37.58942623259064
	Sao Paulo	21	37.4569384034682
		21	37.4569384034682
	Rio de Janeiro	31	37.424211492247906

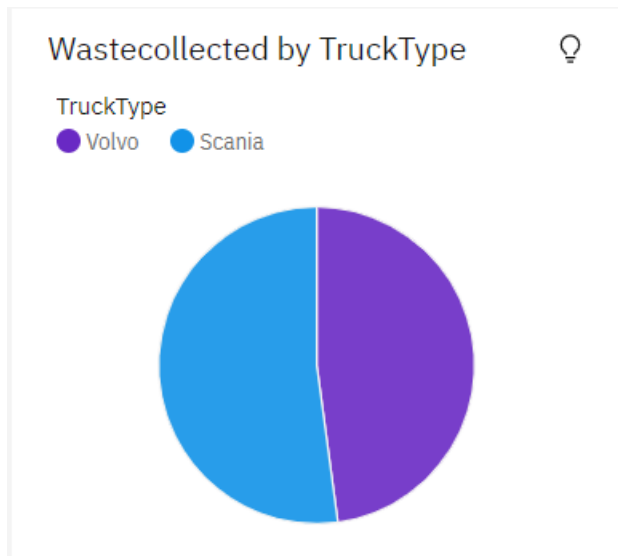
## Task 16 - Create an MQT



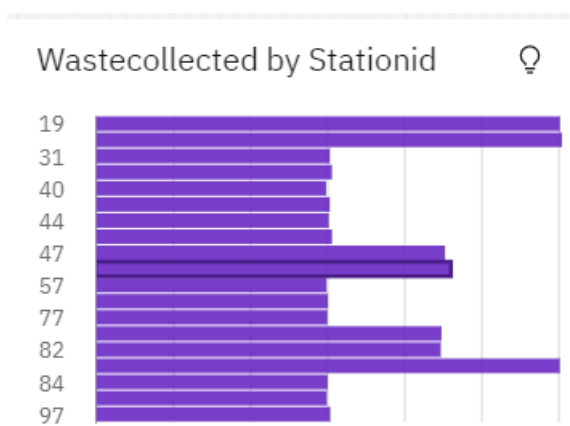
The screenshot shows a SQL IDE with a query editor. The query is a CREATE MATERIALIZED VIEW query that creates a materialized view named maxwastestats1.

```
CREATE TABLE maxwastestats1 (trucktype, stationid, city, max_waste) AS (
  select DT.trucktype, FT.stationid, DS.city, max(FT.wastecollected)
  from facttrips as FT
  left join dimtruck as DT
  on FT.truckid = DT.truckid
  left join dimstation as DS
  on FT.stationid = DS.stationid
  group by (DT.trucktype, FT.stationid, DS.city)
)
DATA INITIALLY DEFERRED
REFRESH DEFERRED
MAINTAINED BY SYSTEM;
```

## Task 17 - Create a pie chart in the dashboard

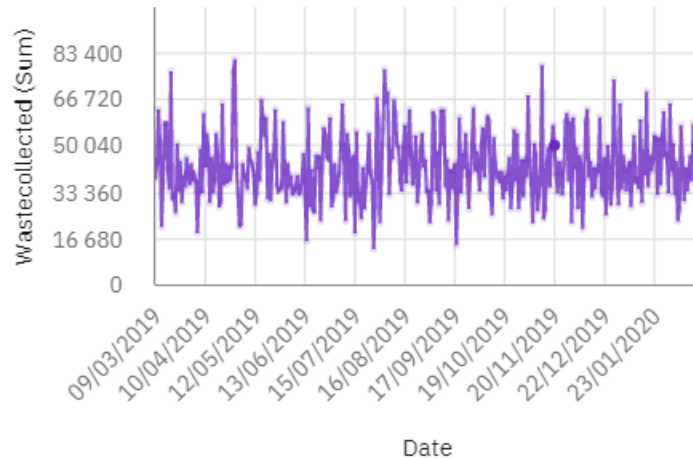


## Task 18 - Create a bar chart in the dashboard



## Task 19 - Create a line chart in the dashboard

Wastecollected by Date



## Task 20 - Create a pie chart in the dashboard

Wastecollected by City



City

● Sao Paulo

● Salvador

● Rio de Janeiro

● Brasilia

