IBM Data Warehousing and Business Intelligence Project

Waste Truck Data Report for Brazilian Cities

By: Hammed Sanni

May 2024

Design a Star Schema Data Warehouse

1. Dimension Tables

DimDate
dateid
date
year
quarter
quartername
month
monthname
day
weekday
weekdayname

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

DimTruck
truckid
trucktype

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

DimStation
stationid
city

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

2. Fact Table

FactTrips
tripid
dateid
stationid
truckid
wastecollected

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

Load Data into Data Warehouse

Task: Create Table in Truck_Waste Database and load Data using PostgreSql

DimDate

i. Create DimDate Table

CREATE TABLE public.DimDate (dateid INTEGER NOT NULL PRIMARY KEY, date DATE NOT NULL,Year INTEGER NOT NULL, Quarter INTEGER NOT NULL, QuarterName VARCHAR(5) NOT NULL, Month INTEGER NOT NULL, Monthname VARCHAR(15) NOT NULL, Day INTEGER NOT NULL, weekday INTEGER NOT NULL, WeekdayName VARCHAR(15) NOT NULL);

ii. Query Top 5 rows

*SELECT * FROM public.DimDate LIMIT 5;*

Data Output | Explain | Messages | Notifications

	dateid [PK] integer	date date	year integer	quarter integer	quartername character varying (5)	month integer	monthname character varying (15)	day integer	weekday integer	weekdayname character varying (15)
1	1	2019-03...	2019	1	Q1	3	March	9	7	Sunday
2	2	2019-03...	2019	1	Q1	3	March	10	1	Monday
3	3	2019-03...	2019	1	Q1	3	March	11	2	Tuesday
4	4	2019-03...	2019	1	Q1	3	March	12	3	Wednesday
5	5	2019-03...	2019	1	Q1	3	March	13	4	Thursday

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

DimTruck

i. Create DimTruck Table

CREATE TABLE public.DimTruck (Truckid INTEGER NOT NULL PRIMARY KEY, TruckType VARCHAR(15) NOT NULL);

ii. Query Top 5 rows

*SELECT * FROM public.DimTruck LIMIT 5;*

Data Output | Explain | Messages | Notifications

	truckid [PK] integer	trucktype character varying (15)
1	115	Volvo
2	120	Scania
3	121	Volvo
4	122	Scania
5	125	Volvo

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

DimStation

i. Create DimStation Table

CREATE TABLE public.DimStation (Stationid INTEGER NOT NULL PRIMARY KEY, city VARCHAR(15) NOT NULL);

ii. Query Top 5 rows

*SELECT * FROM public.DimStation LIMIT 5;*

Data Output | Explain | Messages | Notifications

	stationid [PK] integer	city character varying (15)
1	19	Sao Paulo
2	21	Sao Paulo
3	31	Rio de Janeiro
4	32	Rio de Janeiro
5	40	Brasilia

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

FactTrips

i. Create FactTrips Table

CREATE TABLE public.FactTrips (Tripid INTEGER NOT NULL PRIMARY KEY, Dateid INTEGER NOT NULL, Stationid INTEGER NOT NULL, Truckid INTEGER NOT NULL, Wastecollected double precision NOT NULL,

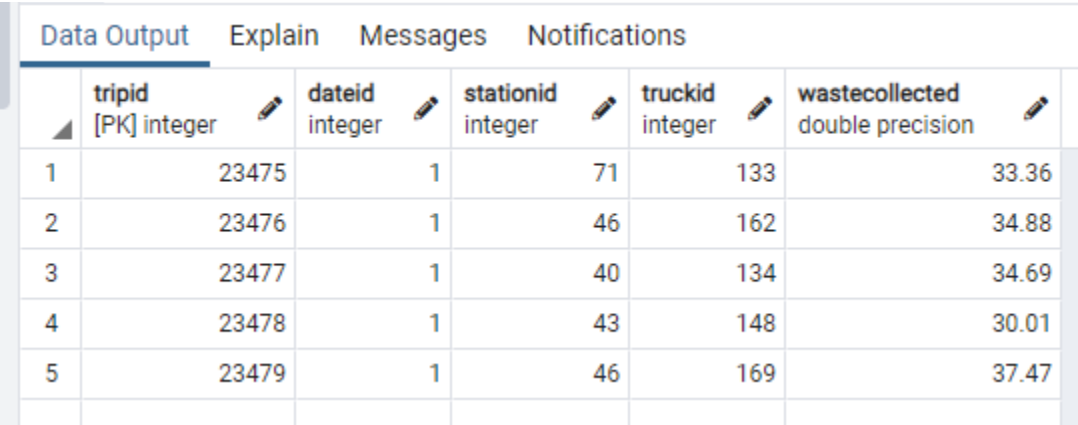
FOREIGN KEY (Dateid) REFERENCES "dimdate" (dateid),

FOREIGN KEY (Stationid) REFERENCES "dimstation" (Stationid),

FOREIGN KEY (Truckid) REFERENCES "dimtruck" (Truckid));

ii. Query Top 5 rows

*SELECT * FROM public.FactTrips LIMIT 5;*



Data Output | Explain | Messages | Notifications

	tripid [PK] integer	dateid integer	stationid integer	truckid integer	wastecollected double precision
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

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

Write Aggregation Queries

i- Grouping Sets SQL

SELECT stationid, trucktype, sum(wastecollected) as totalwastecollected from "facttrips"
left join "dimtruck" on "facttrips".truckid = "dimtruck".truckid
group by grouping sets (stationid, trucktype);

Query Editor | Query History

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

Data Output | Explain | Messages | Notifications

	stationid integer	trucktype character varying (15)	totalwastecollected double precision
1	71	[null]	165131.77000000048
2	84	[null]	166212.16000000038
3	97	[null]	165984.19999999972
4	44	[null]	166456.21999999986
5	82	[null]	246759.30000000005
6	40	[null]	163671.71000000028
7	43	[null]	166730.6400000002
8	48	[null]	254136.8599999996
9	57	[null]	164839.46999999988
10	19	[null]	332816.7800000008
11	81	[null]	247492.7200000002
12	77	[null]	165914.2499999997
13	21	[null]	333142.01000000024
14	83	[null]	332035.3700000003
15	86	[null]	163940.75999999986
16	31	[null]	167061.68

ii- Rollup Query

SELECT year, city, "facttrips".stationid, sum(wastecollected) as totalwastecollected from "facttrips"
left join "dimdate" on "facttrips".dateid = "dimdate".dateid
left join "dimstation" on "facttrips".stationid = "dimstation".stationid
group by rollup (year, city, "facttrips".stationid)
order by year, stationid;

```
1  SELECT year, city, "facttrips".stationid,
2  sum(wastecollected) as totalwastecollected
3  from "facttrips"
4  left join "dimdate"
5  on "facttrips".dateid = "dimdate".dateid
6  left join "dimstation"
7  on "facttrips".stationid = "dimstation".stationid
8  group by rollup (year, city, "facttrips".stationid)
9  order by year, stationid;
10
```

Data Output | Explain | Messages | Notifications

	year integer	city character varying (15)	stationid integer	totalwastecollected double precision
1	2019	Sao Paulo	19	284166.95999999985
2	2019	Sao Paulo	21	283182.3000000004
3	2019	Rio de Janeiro	31	143821.0499999998
4	2019	Rio de Janeiro	32	142766.60000000024
5	2019	Brasilia	40	138850.53999999998
6	2019	Brasilia	43	141503.54000000033
7	2019	Rio de Janeiro	44	141766.3300000001
8	2019	Brasilia	46	144418.7799999999
9	2019	Salvador	47	210905.8499999997
10	2019	Salvador	48	217131.47999999952
11	2019	Rio de Janeiro	57	141405.86
12	2019	Brasilia	71	141856.3800000002
13	2019	Brasilia	77	141627.09999999963

iii- Cube Query

SELECT year, city, "facttrips".stationid, avg(wastecollected) as averagewastecollected from "facttrips"
left join "dimdate" on "facttrips".dateid = "dimdate".dateid
left join "dimstation" on "facttrips".stationid = "dimstation".stationid
group by cube (year, city, "facttrips".stationid)
order by year, stationid;

```
SELECT year, city, "facttrips".stationid,
avg(wastecollected) as averagewastecollected
from "facttrips"
left join "dimdate"
on "facttrips".dateid = "dimdate".dateid
left join "dimstation"
on "facttrips".stationid = "dimstation".stationid
group by cube (year, city, "facttrips".stationid)
order by year, stationid;
```

Data Output | Explain | Messages | Notifications

	year integer	city character varying (15)	stationid integer	averagewastecollected double precision
1	2019	Sao Paulo	19	37.56834479111579
2	2019	[null]	19	37.56834479111579
3	2019	Sao Paulo	21	37.443117810392756
4	2019	[null]	21	37.443117810392756
5	2019	[null]	31	37.463154467309145
6	2019	Rio de Janeiro	31	37.463154467309145
7	2019	[null]	32	37.520788436268134
8	2019	Rio de Janeiro	32	37.520788436268134
9	2019	Brasilia	40	37.37565006729474
10	2019	[null]	40	37.37565006729474
11	2019	[null]	43	37.504251258945224
12	2019	Brasilia	43	37.504251258945224
13	2019	Rio de Janeiro	44	37.454776750330275

Create Materialized View

CREATE MATERIALIZED VIEW max_waste_stats (city, stationid, trucktype, maxwastecollected) AS
(select city, "facttrips".stationid, trucktype, max(wastecollected) FROM "facttrips"
left join "dimstation" on "facttrips".stationid = "dimstation".stationid
left join "dimtruck" on "facttrips".truckid = "dimtruck".truckid
group by city, "facttrips".stationid, trucktype);

```
CREATE MATERIALIZED VIEW max_waste_stats
(city, stationid, trucktype, maxwastecollected)
AS (select city, "facttrips".stationid, trucktype, max(wastecollected)
FROM "facttrips"
left join "dimstation"
on "facttrips".stationid = "dimstation".stationid
left join "dimtruck"
on "facttrips".truckid = "dimtruck".truckid
group by city, "facttrips".stationid, trucktype);
```

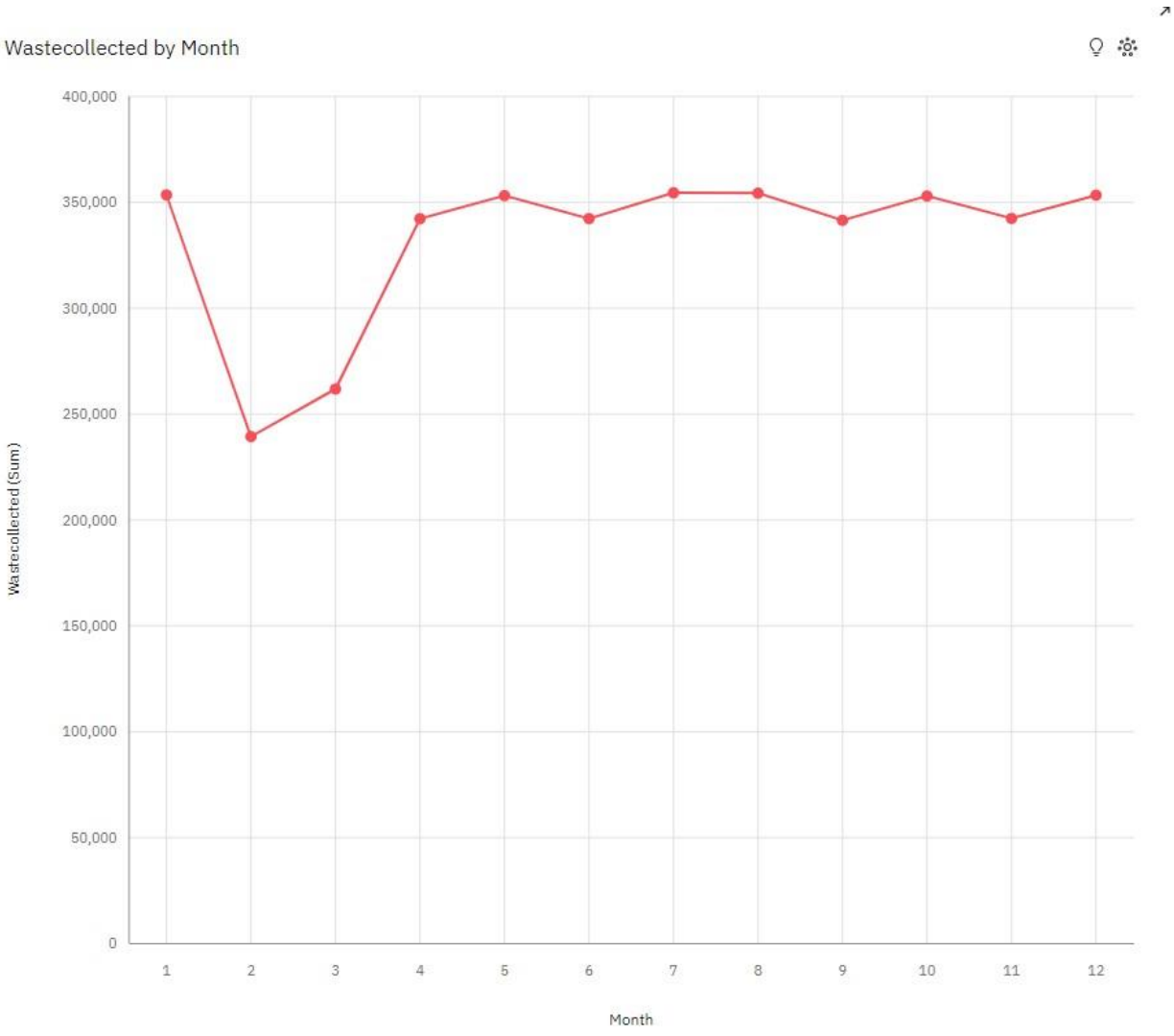
```
1  select * from max_waste_stats;
```

Data Output | Explain | Messages | Notifications

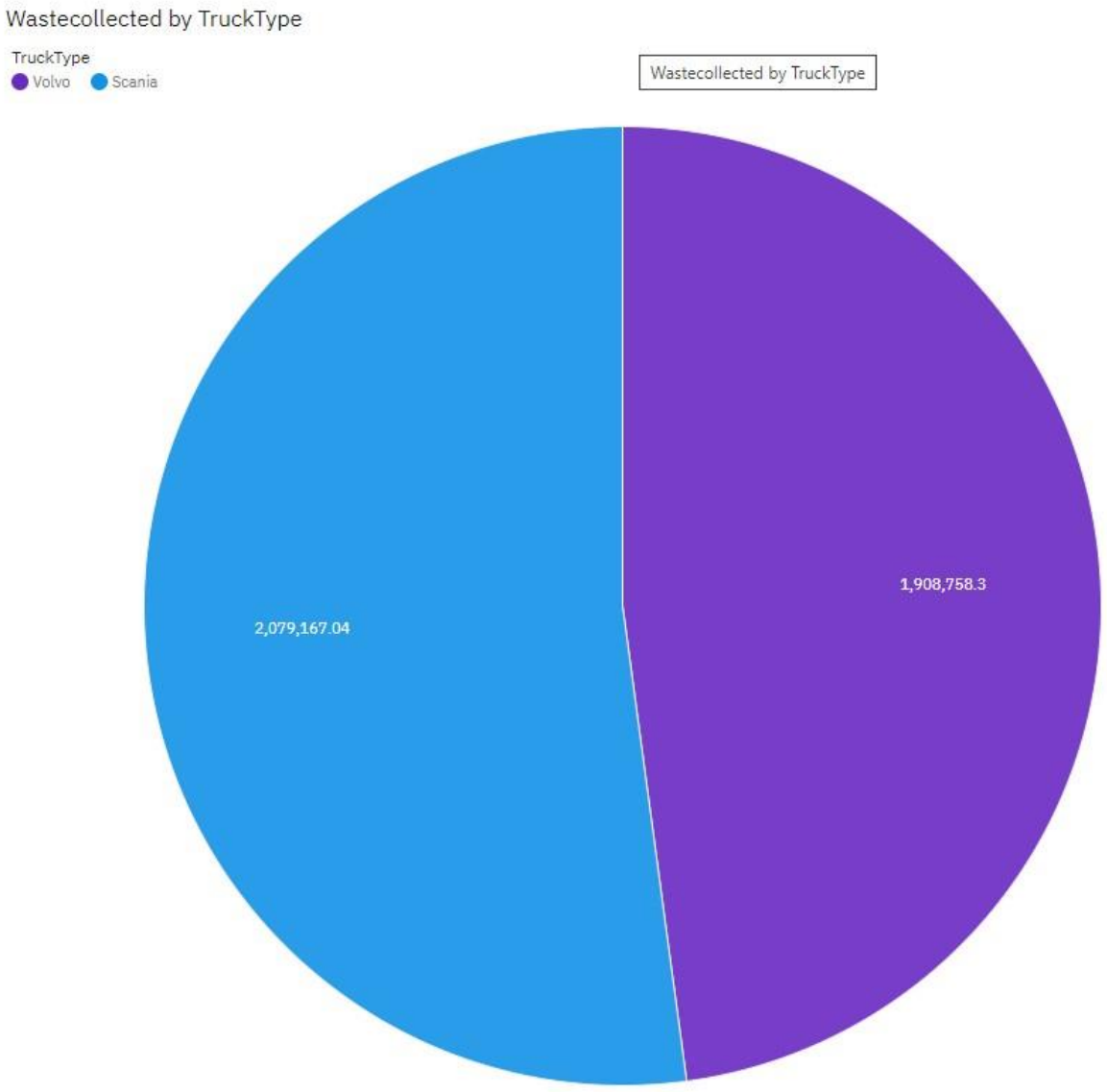
	city character varying (15)	stationid integer	trucktype character varying (15)	maxwastecollected double precision
1	Rio de Janeiro	32	Volvo	44.99
2	Brasilia	77	Scania	45
3	Brasilia	97	Volvo	45
4	Rio de Janeiro	31	Scania	45
5	Rio de Janeiro	31	Volvo	45
6	Brasilia	43	Scania	45
7	Salvador	47	Scania	45
8	Sao Paulo	21	Scania	45
9	Rio de Janeiro	84	Scania	45
10	Brasilia	46	Volvo	45
11	Brasilia	40	Scania	45
12	Sao Paulo	19	Volvo	45
13	Rio de Janeiro	44	Volvo	44.99
14	Rio de Janeiro	86	Scania	45
15	Brasilia	77	Volvo	44.98
16	Salvador	81	Scania	45
17	Salvador	82	Volvo	45

Create Dashboard using IBM Cognos Analytics

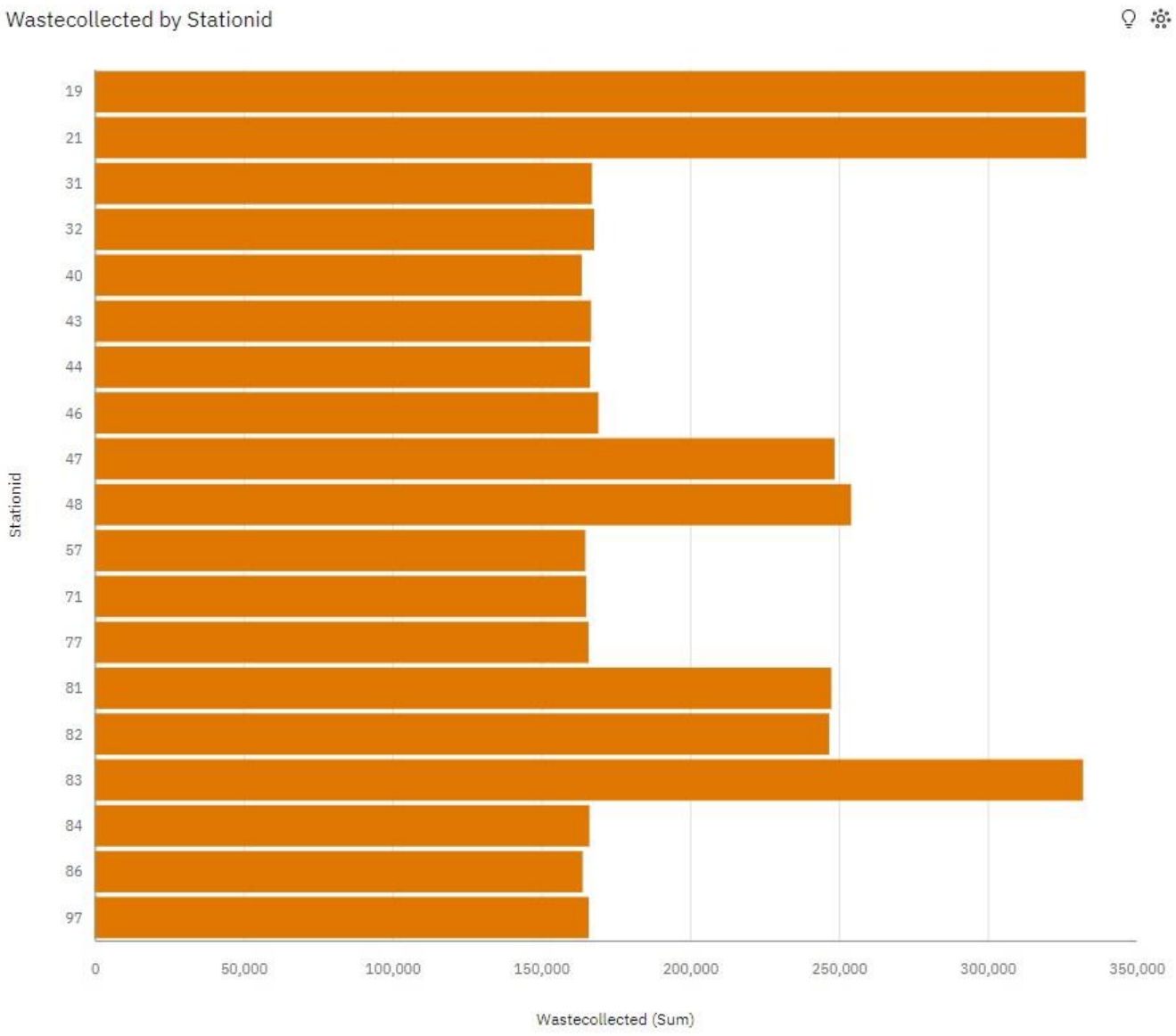
i. Total waste collected per Month



ii. Total waste collected by Truck Type



iii. Waste Collected by Station



iv. Waste Collected by City