Public Transport Coverage in Metropolitan Melbourne

Name: Jeet Faldu

Contents

1. Introduction

Transition from remote to face-to-face work in Victoria after Covid-19 has resulted in a rise in demand for public transport. PTV responsible for Victoria's public transport network has restored their services to cover as many areas as possible in the whole area. However, there are multiple questions raised about the restored services like **coverage** of transport network, where there are **gaps**, and which areas provide the **best transit** options. Mainly two datasets **GTFS** (General Transit Feed Specification) which provides transit data and **Australian digital boundary** from the Australian Statistical Geography Standard which provides Australia's geographic divisions. This analysis provides answers to questions raised with the restored services using the above-mentioned datasets.

2. Methodology

Dataset Overview

Two key datasets used for the analysis are GTFS dataset and Australian boundary data.

I. GTFS: The General Transit Feed Specification (GTFS) is a format for agencies to publish their transportation data in a specific manner. It contains information like transit routes, trips along with travel time, stops and calendar with services available on different days. For this analysis 17th March 2023 PTV GTFS dataset is being used. Further information about data can be found on the GTFS Page.

II. Australian Boundary Data: This dataset is derived from Australian Bureau of Statistics which provides geographic design across country. For this analysis, along with ABS structure non-ABS structure will be used. ABS structures provide mesh blocks which are created by considering 30-60 building blocks and are smallest geographic areas. Non-ABS structure includes information about administration regions. More information about this dataset is available at link.

Data Restoration

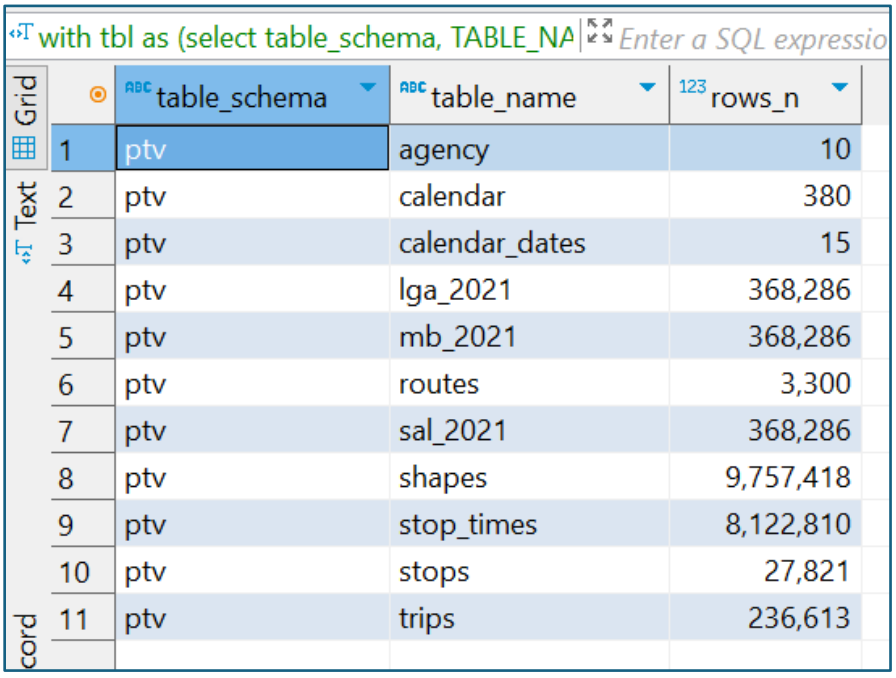
For data restoration process below procedure was followed:

- Using DBeaver created schema 'ptv' for the dataset to add tables for the analysis and store relevant data in the schema.

- Data set stored in archived format were unzipped using ‘unzip’ utility in Docker container terminal. Additionally, it was observed that two files in GTFS dataset and two files of Mesh Block dataset were archived.
- It was observed that raw data files were in three different file formats shp(shape), csv for Australian boundary data and txt files for GTFS dataset. Further, two methodologies were used to handle different file formats. One for Shape and CSV files and one for text files.
- The efficient way to create tables for CSV and shape files was to use ‘OGR2OGR’ which converts simple features data between file formats. CLI for PostgreSQL is used to connect to database and create table. Additionally, -nlt multipolygon was used to restore the multipolygon shapes. Using this method tables were created for the shape and CSV data files for Australian boundary dataset.
- For text files, DDL scripts were created for all the 8 GTFS data files. Table structures were created using DDL scripts. Further, data were copied in the table using ‘COPY’ command in which path of the data files were specified along with the delimiter and flag for the header in the dataset.

Data Verification

- Using procedures defined in the data restoration process 11 tables were created in the PTV schema which includes 8 GTFS tables and 3 Australian Boundary data tables.
- Additional information about all the 11 tables can be found in the screenshot attached below which indicates tables are created successfully. (Query in Appendix-1)



	table_schema	table_name	rows_n
1	ptv	agency	10
2	ptv	calendar	380
3	ptv	calendar_dates	15
4	ptv	lga_2021	368,286
5	ptv	mb_2021	368,286
6	ptv	routes	3,300
7	ptv	sal_2021	368,286
8	ptv	shapes	9,757,418
9	ptv	stop_times	8,122,810
10	ptv	stops	27,821
11	ptv	trips	236,613

Figure 1

Data Preprocessing

Australian Boundary Data:

For this analysis the focus was on the Melbourne Metropolitan area. Since mesh blocks contains whole Australia geography boundary it needs to be filtered for Melbourne Metropolitan area. Below procedure was followed to filter boundary data for Melbourne region.

- Mesh Block Table: It was identified that GCC column which indicates Greater Capital City which are built from Statistical Areas Level 4 (SA4s) can be used to filter mesh blocks for Melbourne Region. Additionally, it was observed these mesh blocks can be identified using 'Greater Melbourne' which contains blocks within Melbourne and surrounding areas. A new table 'mb2021_mel' was created for mesh blocks within the Melbourne region to reduce the cost of queries. It was observed that the number of records dropped from 368k to 59K after applying filter for 'Greater Melbourne'. (Query in Appendix-2.1)

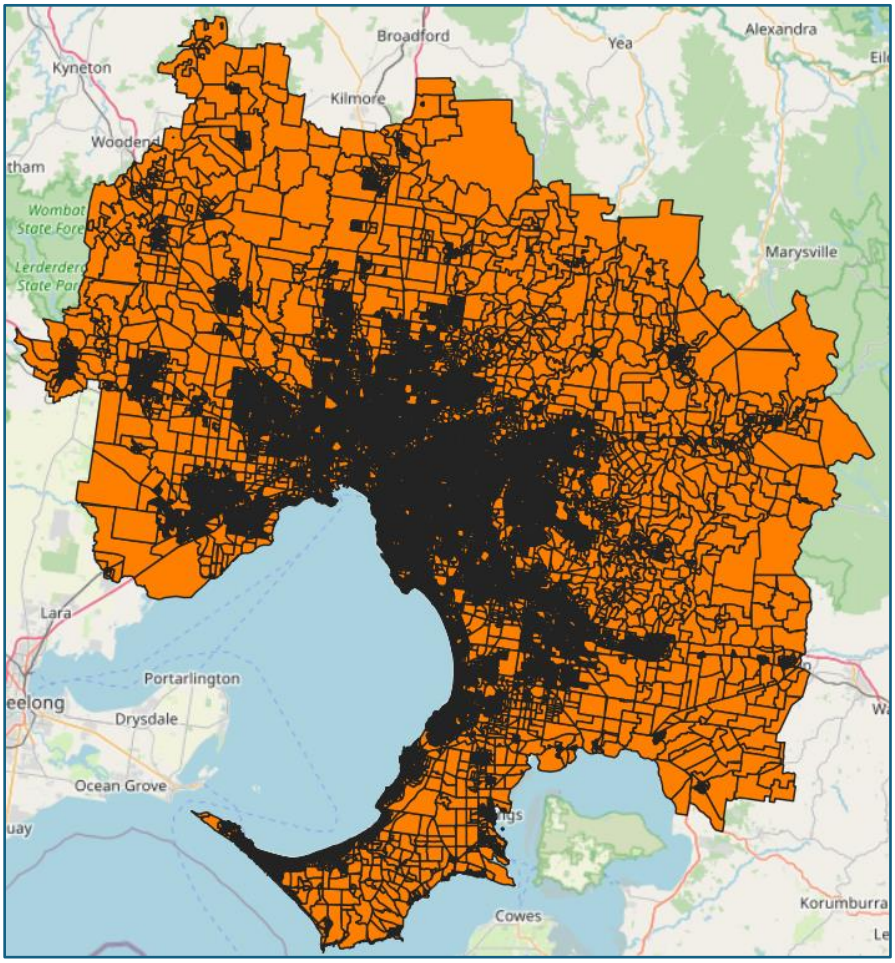


Figure 2

- SAL and LGA Table: Like mesh block table these SAL and LGA tables also contains information about whole Australia whereas requirement for this analysis narrow downs to Melbourne Region. To filter region for Melbourne in these tables, new tables were created 'lga_2021_mel_geom' and 'sal_2021_mel_geom' using 'mb2021_mel' to join on 'mb_code_2021'. On filtering these tables, it was observed that records dropped from 368K to 59K. Additionally, geometry column was included in these newly created tables for further analysis based on LGA and SAL. (Query in Appendix-2.2)
- Mesh Block Aggregation: It was observed for further analysis along with individual mesh blocks aggregated polygon of Melbourne will be required. Using ST Union function all the mesh blocks for Melbourne region were combined and stored in table 'melbourne_metropolitan'. (Query in Appendix-2.3)

Figure 3

GTFS Dataset

For this dataset manipulation is performed for individual tables and new tables were created after integrating relevant information. Below manipulation were performed on GTFS dataset:

- Geometry column was not present in the Stops column. However, latitude and longitude were available for each stop. Using 'AddGeometryColumn' function new column was introduced in the stops table. Further, latitude and longitude were converted to geom point using Set SRID function along with Make point. (Query in Appendix-2.4)
- Further it was observed that required information like vehicle type, trip Id, route name etc. was not present in the stops table. However, this information was present in other tables like routes and trips. New table 'stops_routes' was created by collecting all the relevant information from Stops, Routes and Trips tables. Additionally, information for

vehicles was not directly provided in the dataset but route type indicates the corresponding vehicle type. A vehicle type column was created in this new table considering 0 corresponds to tram, 2 to train and 3 to bus. (Query in Appendix-2.5)

Data Analysis and Visualization

The focus of the analysis was on regions with high numbers of offices as post-covid offices are switching from work from to face-to-face mode. In Melbourne most offices are in the CBD area which was taken into consideration but most people commute from surrounding regions to CBD area. Hence, along with Melbourne LGA, six LGAs sharing boundary with Melbourne LGA were taken into consideration.

To analyze the best area in Melbourne Metropolitan I initially investigated gaps in Melbourne region at LGA (Local Government Authority) and SAL (Suburbs and Localities) levels. On investigation it was observed that there is gap in one LGA out of total 36 LGA in greater Melbourne without any stops. However, on visualizing the LGA on map it was observed that missing LGA 'Unincorporated Vic' is an island, and it is expected that connectivity of public transport is unavailable on an island. Similarly, for suburb level it was observed that 115 suburbs don't have any stops out of 577 suburbs in Greater Melbourne. However, most of these suburbs lie on the outskirts of Melbourne region which might be under development and there is not much requirement of public transport according to current scenario. (Query in Appendix-3.1)

Figure 4 LGA Gap

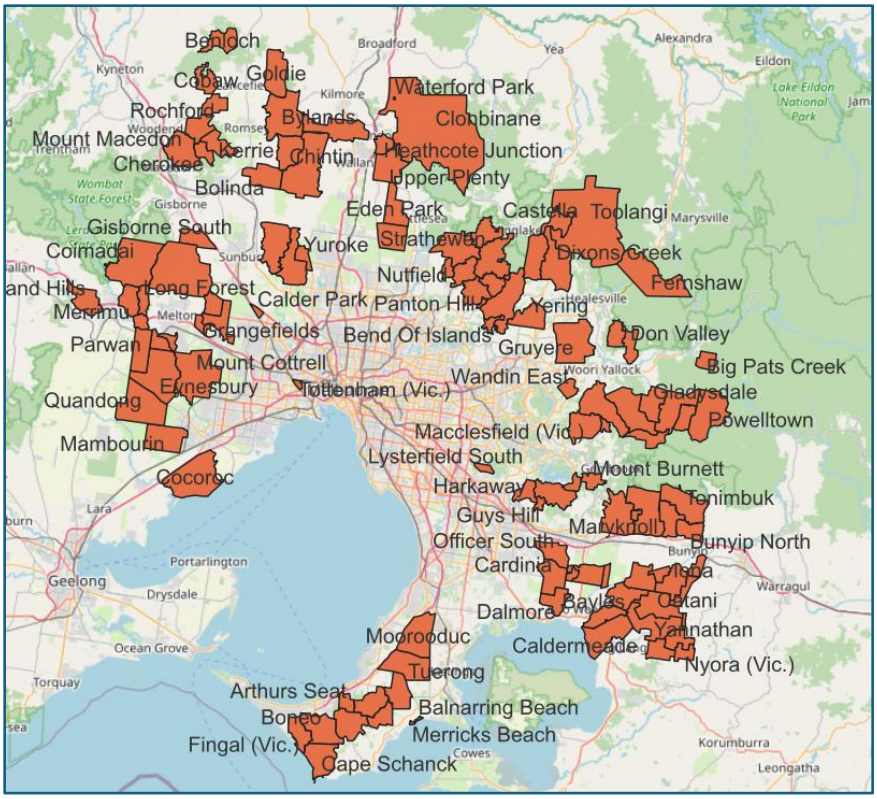


Figure 5 SAL GAP

Further part of analysis I explored coverage of public transport in Melbourne Region for LGA and SAL levels. For LGA it was observed that region surrounding Melbourne CBD region has either High or medium coverage which signifies that coverage of public transport around city is high as expected. LGA away from city region have either low coverage or very low coverage again this is expected as these regions are either developing or have less population which doesn't require very high connectivity of public transport. Similar trends regarding public transport coverage were observed at SAL level where High and Medium coverage is observed for most SAL in center part of Melbourne region and very low and low coverage on outskirts of Melbourne region. (Query in Appendix-3.2)

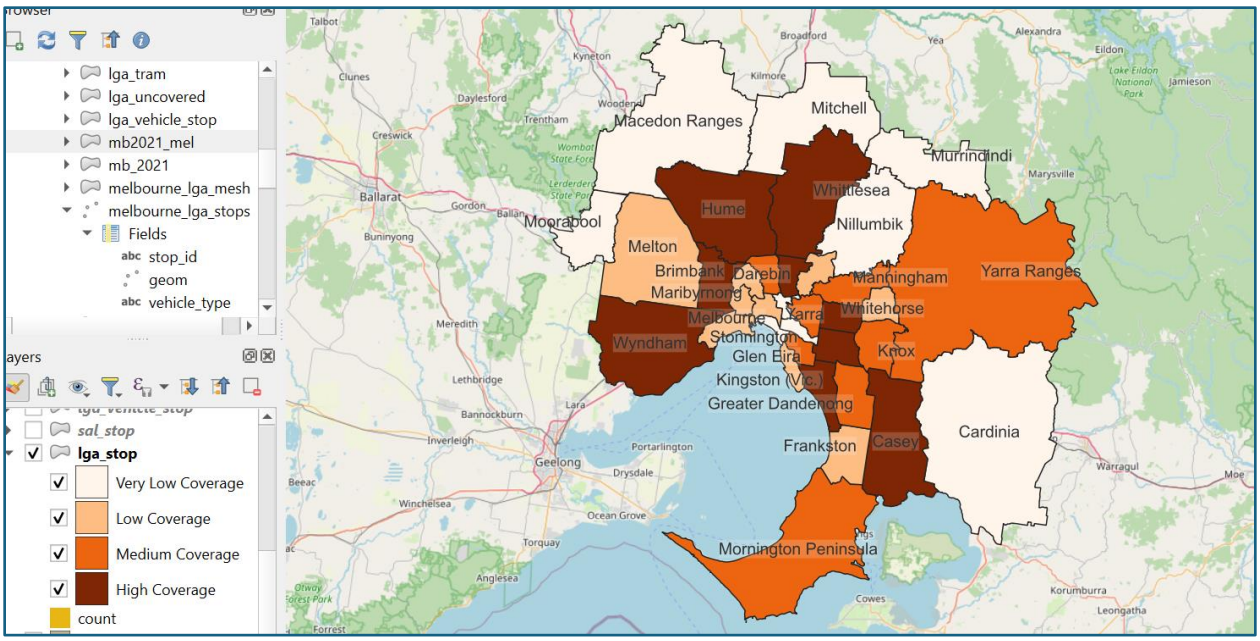


Figure 6 LGA Coverage

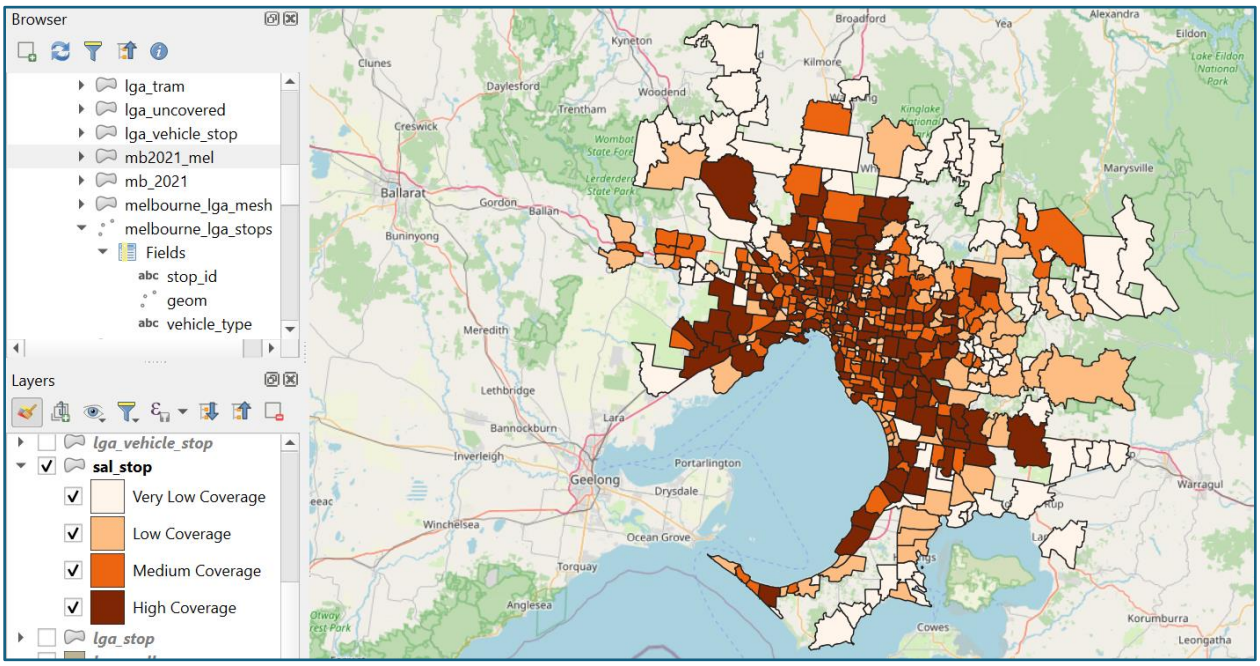


Figure 7 SAL Coverage

Exploration of gaps and coverage in Melbourne region were evident enough to support the analysis and led to further exploration about best area in Melbourne Metropolitan region. To identify the best area, instead of focusing on only stop counts I split investigation based on vehicle type. Additionally, to consider area factors for each region ratio of number of stops to area in square kilometers is taken. Based on vehicle type exploration it was observed that for the top 5 LGA with highest ratio between number of stops and area square kilometers different LGAs were

dominating for different vehicle type. Though, it was observed that all three categories dominating LGA were surrounded by Melbourne LGA. However, the dominating SAL based on vehicle type showed little different trends than LGA as they were not only surrounded by Melbourne SAL but little away from the city region as well. (Query in Appendix-3.3)

LGA Visuals:

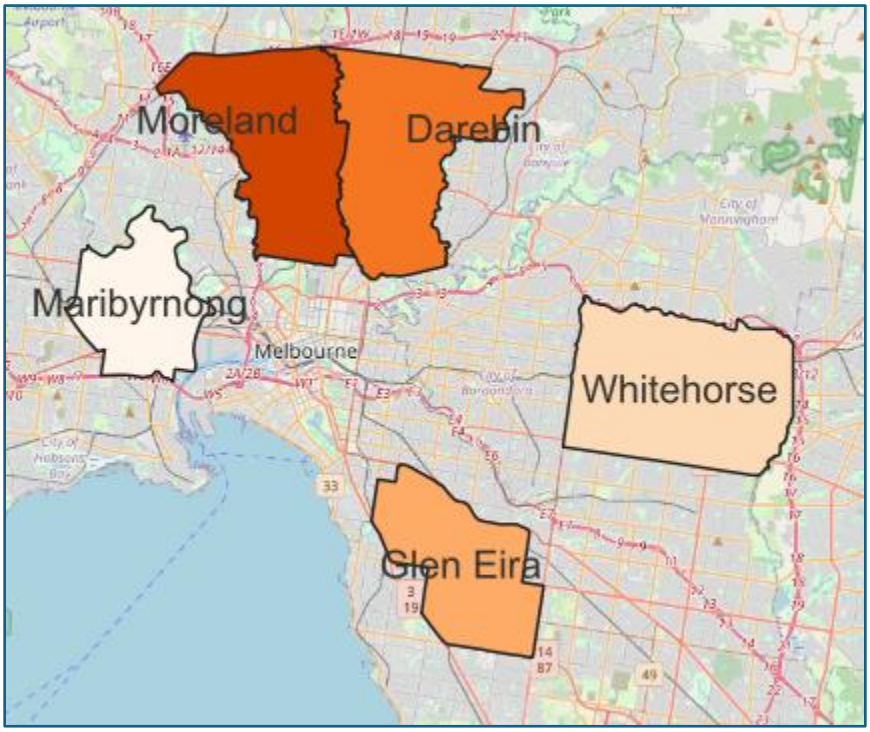


Figure 8 LGA Bus

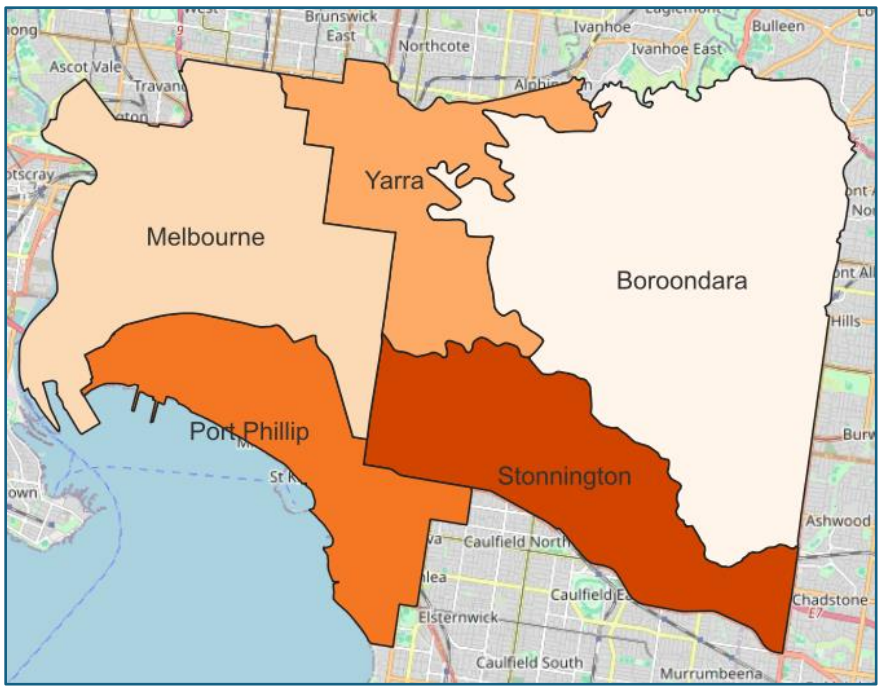


Figure 9 LGA Tram

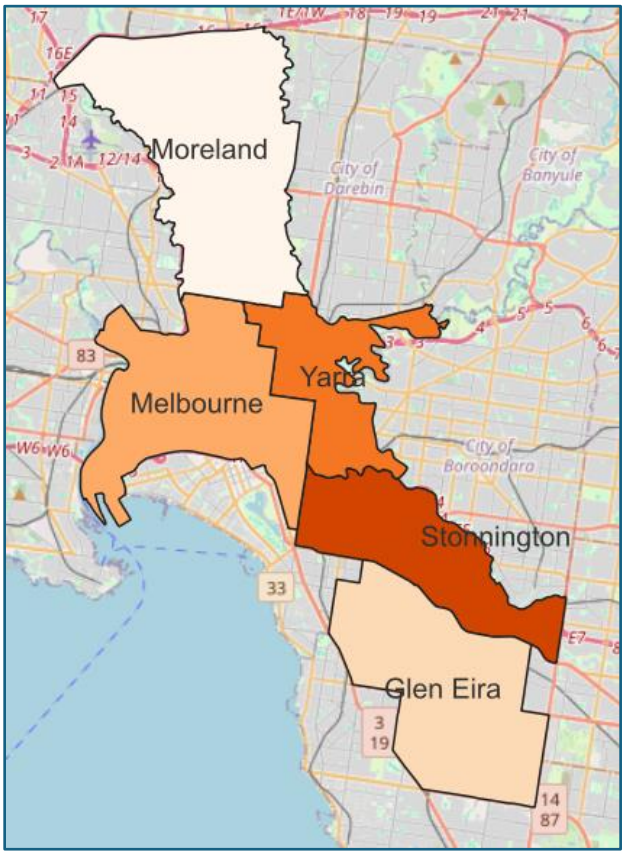


Figure 10 LGA Train

SAL Visuals:

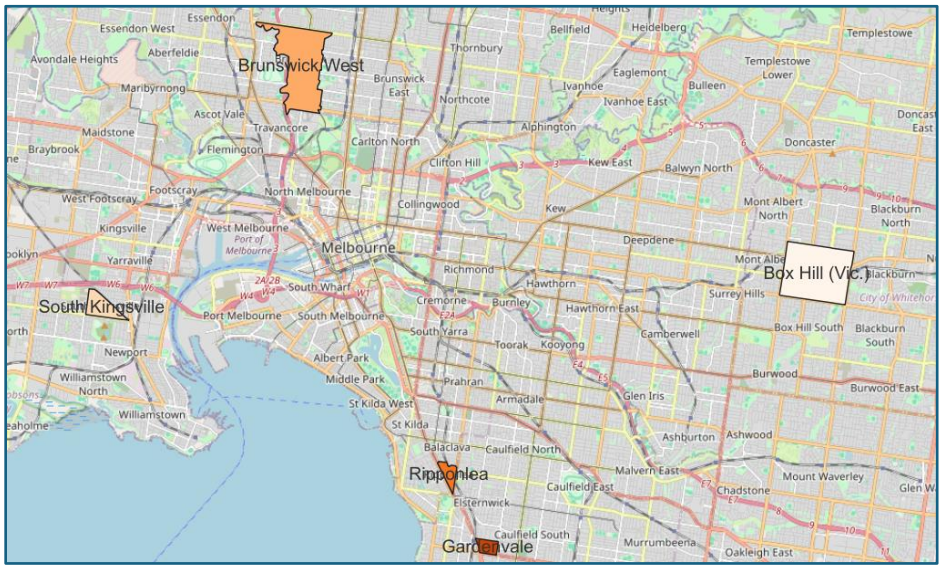


Figure 11 SAL Bus

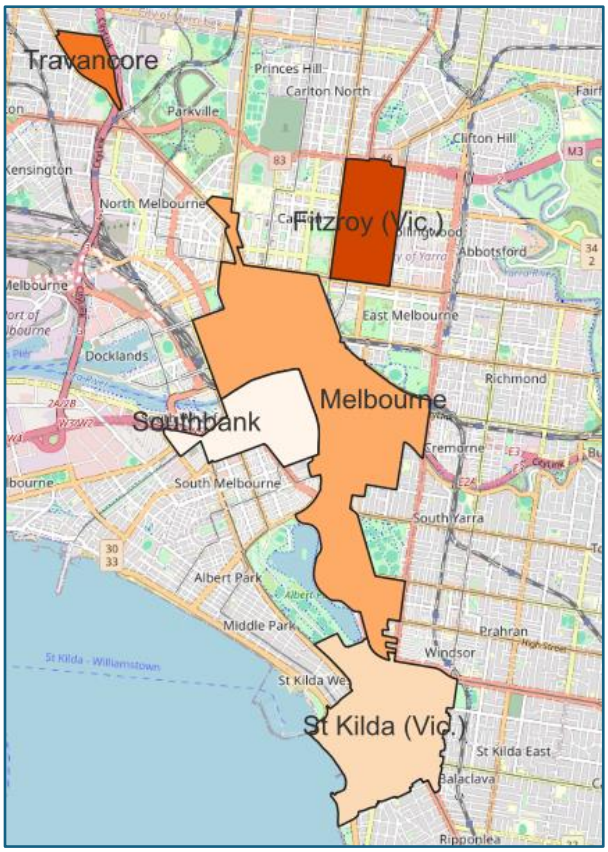


Figure 12 SAL Tram

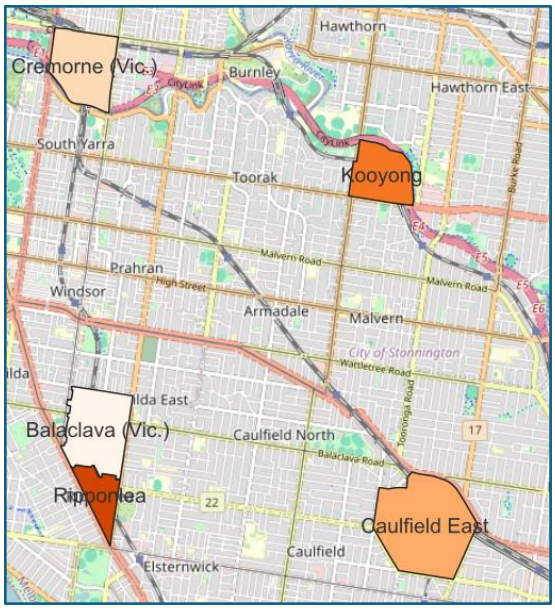


Figure 13 SAL Train

Finally, to explore Melbourne and surrounding LGAs considered the stop points based on vehicle type and mesh blocks for each LGA. Based on visuals it can be considered that bus coverage is good in all 7 LGAs but it is moderate in Melbourne, Port Phillip and Stonnington. However, for train and tram these three LGAs Melbourne, Port Phillip and Stonnington have bet network in comparison to other 4 LGAs MORELAND, MOONEE VALLEY, MARIBYRNONG, and HOBSONS BAY. According to the location of these LGAs, ones with highly congested regions, it is expected that the network of train and tram should be best as it is efficient to have train or tram in place of buses. On the other hand, less congested regions should have better networks for buses, and it is evident from the visuals that those regions have better connectivity of buses. (Query in Appendix-3.4)

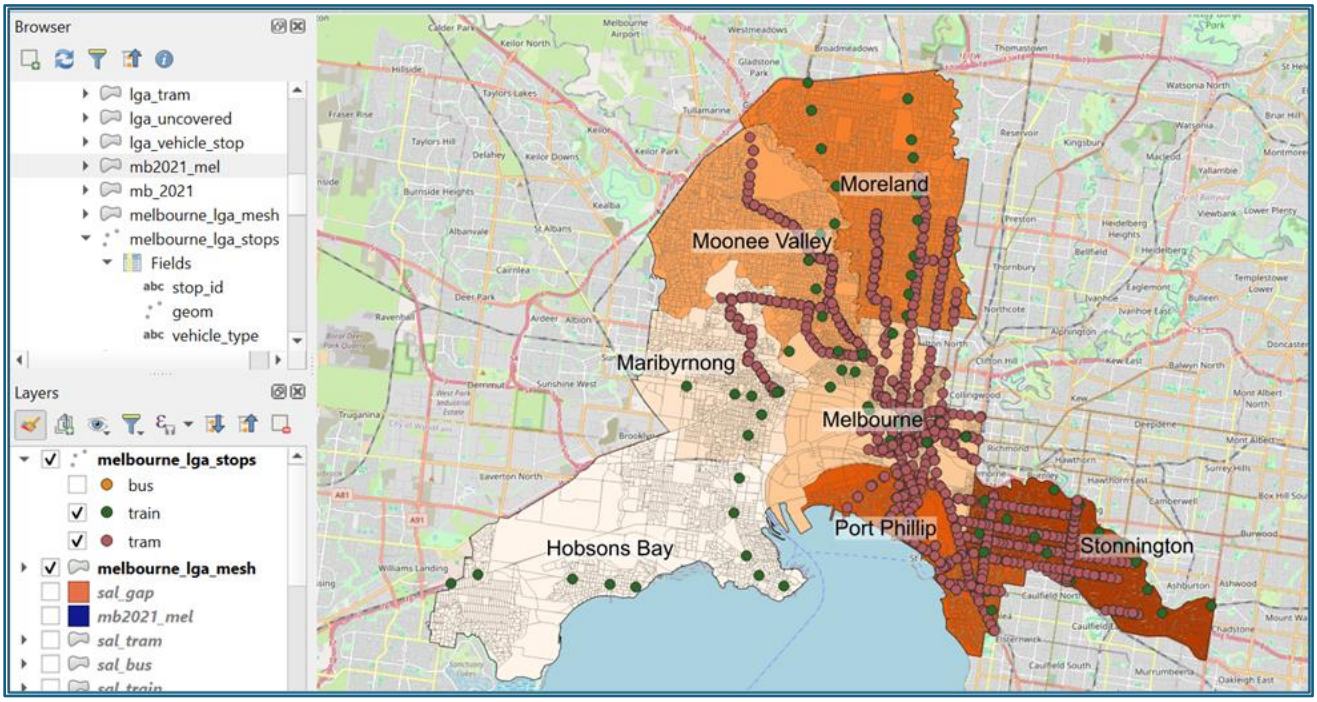


Figure 14 Tram/Train

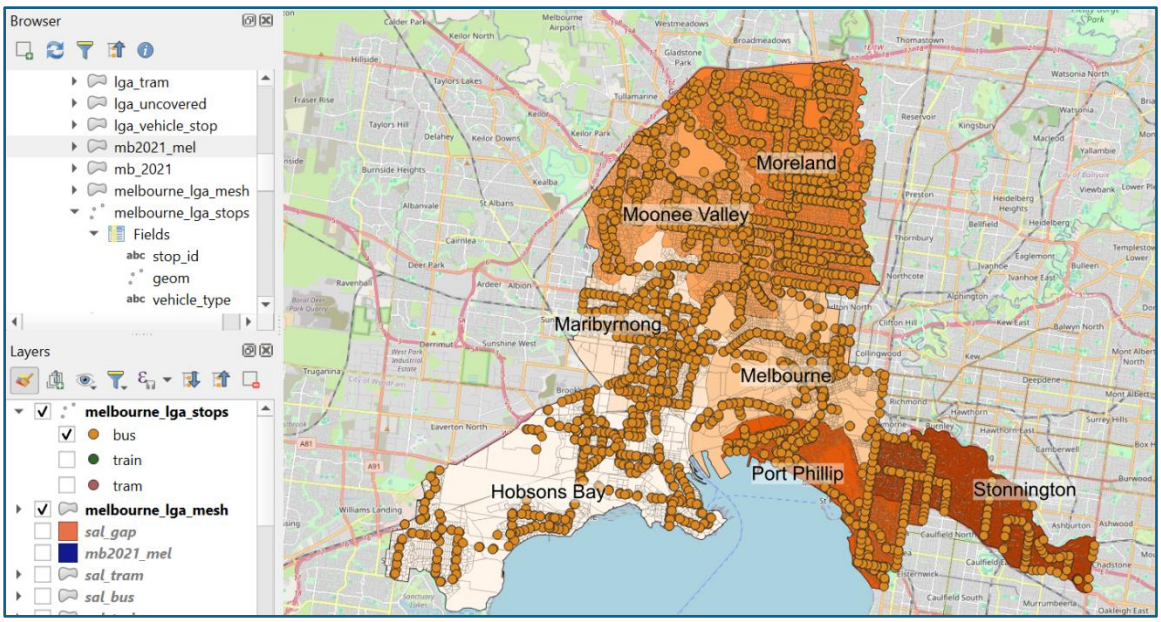


Figure 15 Bus

Software and libraries used for the above analysis are mentioned below:

- QGIS: To visualize the spatial data on Map all visuals were created using QGIS software.
- SQL: Manipulation of data was done using SQL
- PostgreSQL: Manipulated data was stored using PostgreSQL relational database
- DBeaver: Database tool used for writing SQL queries and connecting to database was done using DBeaver.

3. Results

On investigating the datasets, it was observed that for both LGAs and SALs there were no regions near Melbourne CBD without any stops. However, there were a few regions which didn't have any stops, but such regions were either an island or far from city region, which is expected as these regions might be developing, and public transport is not expected in these regions.

The same trend was observed for coverage of public transport where coverage is better near city region, but network of public transport weakens as region moves away from city. However, for 'Tottehnam' suburb, it was observed that it is the only suburb near CBD region without any stops, but it has stop in neighboring suburb which gives restructuring opportunity to PTV.

On exploring Melbourne region for vehicle type, it was observed that connectivity of all vehicle types i.e. train, tram and bus are not dominated by any region. However, regions which have a high population density have better connectivity of train or trams. On the other hand, regions with less congestion have better connectivity of bus network.

Henceforth, overall connectivity of Melbourne region which consists of many offices may get impacted by uplift of covid-19 restriction, but with a little improvement, current network of public transport is sufficient to handle rise in demand of public transport.

4. Discussion

Gaps: There was no gap in the LGA and SAL regions as such but there were very few regions without any stops which were either island or outside of Melbourne CBD region.

Coverage: Public transport network is better in CBD region and its surroundings. However, network coverage weakens as regions move away from CBD region.

Best Public Transportation: Considering vehicle type different regions have different networks. Regions with high population density have better services for train and trams. Whereas low density regions have better networks of bus services. Additionally, the connectivity of public transport near the Melbourne region is better with either vehicle type which can cater to rise in demand of public transport post covid-19.

Limitation: Current analysis consists of GTFS dataset from 17 March 2023 because of which analyses are restricted to current time frame and any change in PTV network can't be analyzed with given dataset.

5. References

6. Appendix

1. Data Restoration

Schema Creation:

```
create schema ptv;
```

Unzip utility installation:

```
PS C:\Users\jeetf> docker exec -it 16a5df225e67 bash
root@16a5df225e67:/home/student# apt update && apt install unzip
```

Unzipping data:

```
root@16a5df225e67:/home/student# unzip /data/adata/MB_2021_AUST_SHP_GDA2020/MB_2021_AUST_GDA2020.shp.zip -d /data/adata/MB_2021_AUST_SHP_GDA2020/
```

```
root@16a5df225e67:/home/student# unzip /data/adata/MB_2021_AUST_SHP_GDA2020/MB_2021_AUST_GDA2020.shp.zip -d /data/adata/MB_2021_AUST_SHP_GDA2020/
```

```
root@16a5df225e67:/home/student# unzip /data/adata/gtfs/shapes.txt.zip -d /data/adata/gtfs/
```

```
root@16a5df225e67:/home/student# unzip /data/adata/gtfs/stop_times.txt.zip -d /data/adata/gtfs/
```

Table Creation using Ogr2Ogr:

```
root@16a5df225e67:/home/student# ogr2ogr PG:"dbname=gisdb user=postgres" "/data/adata/MB_2021_AUST_SHP_GDA2020/MB_2021_AUST_GDA2020.shp" -nln ptv.mb_2021 -overwrite -nlt MULTIPOLYGON
```

```
root@16a5df225e67:/home/student# ogr2ogr PG:"dbname=gisdb user=postgres" "/data/adata/LGA_2021_AUST.csv" -nln ptv.LGA_2021 -overwrite
```

```
root@16a5df225e67:/home/student# ogr2ogr PG:"dbname=gisdb user=postgres" "/data/adata/SAL_2021_AUST.csv" -nln ptv.SAL_2021 -overwrite
```

Table Creation using DDL and copying data:

```
CREATE TABLE ptv.agency(
 agency_id numeric,
 agency_name varchar,
 agency_url varchar,
 agency_timezone varchar,
 agency_lang  varchar);

-- Load data using COPY method
 COPY ptv.agency(agency_id,agency_name,agency_url,agency_timezone,agency_lang)
 FROM '/data/adata/gtfs/agency.txt'
 delimiter ','
 csv header;

 select * from ptv.agency;
```

```sql
CREATE TABLE ptv.calendar (
    service_id VARCHAR PRIMARY KEY,
    monday numeric,
    tuesday numeric,
    wednesday numeric,
    thursday numeric,
    friday numeric,
    saturday numeric,
    sunday numeric,
    start_date DATE,
    end_date DATE
);

-- Load data using COPY method
COPY ptv.calendar(service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date)
FROM '/data/adata/gtfs/calendar.txt'
delimiter ','
csv header;

select * from ptv.calendar;
```

```sql
CREATE TABLE ptv.calendar_dates (
    service_id VARCHAR,
    date DATE,
    exception_type numeric
);

-- Load data using COPY method
COPY ptv.calendar_dates(service_id,date,exception_type)
FROM '/data/adata/gtfs/calendar_dates.txt'
delimiter ','
csv header;


select * from ptv.calendar_dates;
```

```sql
CREATE TABLE ptv.routes(
    route_id VARCHAR,
    agency_id VARCHAR,
    route_short_name VARCHAR,
    route_long_name VARCHAR,
    route_type VARCHAR,
    route_color VARCHAR,
    route_text_color VARCHAR
);

-- Load data using COPY method
COPY ptv.routes(route_id,agency_id,route_short_name,route_long_name,route_type,route_color,route_text_color)
FROM '/data/adata/gtfs/routes.txt'
delimiter ','
csv header;


select distinct route_type from ptv.routes;
```

```sql
CREATE TABLE ptv.stops(
    stop_id VARCHAR,
    stop_name VARCHAR,
    stop_lat FLOAT,
    stop_lon FLOAT
);

-- Load data using COPY method
COPY ptv.stops(stop_id,stop_name,stop_lat,stop_lon)
FROM '/data/adata/gtfs/stops.txt'
delimiter ','
csv header;


select * from ptv.stops;
```

```sql
CREATE TABLE ptv.trips(
    route_id VARCHAR,
    service_id VARCHAR,
    trip_id VARCHAR,
    shape_id VARCHAR,
    trip_headsign VARCHAR,
    direction_id NUMERIC
);

-- Load data using COPY method
COPY ptv.trips(route_id,service_id,trip_id,shape_id,trip_headsign,direction_id)
FROM '/data/adata/gtfs/trips.txt'
delimiter ','
csv header;


select * from ptv.trips;
```

```sql
CREATE TABLE ptv.shapes(
    shape_id VARCHAR,
    shape_pt_lat FLOAT,
    shape_pt_lon FLOAT,
    shape_pt_sequence NUMERIC,
    shape_dist_traveled FLOAT
);

-- Load data using COPY method
COPY ptv.shapes(shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence,shape_dist_traveled)
FROM '/data/adata/gtfs/shapes.txt'
delimiter ','
csv header;


select * from ptv.shapes;
```

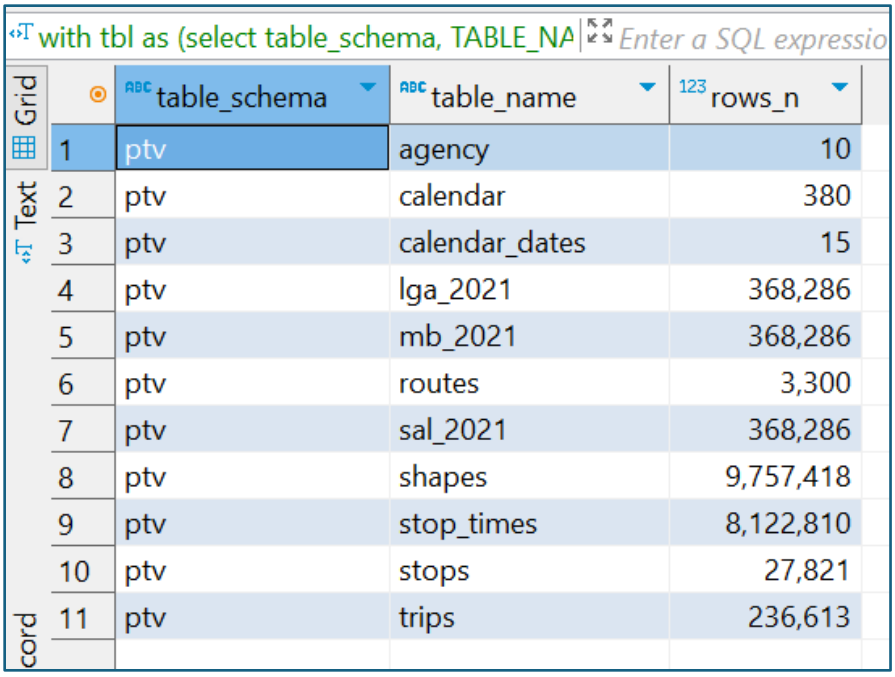
```sql
CREATE TABLE ptv.stop_times(
    trip_id VARCHAR,
    arrival_time VARCHAR,
    departure_time VARCHAR,
    stop_id VARCHAR,
    stop_sequence NUMERIC,
    stop_headsign VARCHAR,
    pickup_type NUMERIC,
    drop_off_type NUMERIC,
    shape_dist_traveled VARCHAR
);

-- Load data using COPY method
COPY ptv.stop_times(trip_id,arrival_time,departure_time,stop_id,stop_sequence,stop_headsign,pickup_type,
drop_off_type,shape_dist_traveled)
FROM '/data/adata/gtfs/stop_times.txt'
delimiter ','
csv header;


select * from ptv.stop_times;
```

Data Verification

```sql
with tbl as
 (select table_schema, TABLE_NAME
  from information_schema.tables
  where table_schema in ('ptv'))
select table_schema, TABLE_NAME,
(xpath('/row/c/text()', query_to_xml(format('select count(*) as c from %I.%I', table_schema, TABLE_NAME), FALSE, TRUE, ''))
from tbl
order by table_name;
```



with tbl as (select table_schema, TABLE_NA | Enter a SQL expressio

Grid | Text | cord

	table_schema	table_name	rows_n
1	ptv	agency	10
2	ptv	calendar	380
3	ptv	calendar_dates	15
4	ptv	lga_2021	368,286
5	ptv	mb_2021	368,286
6	ptv	routes	3,300
7	ptv	sal_2021	368,286
8	ptv	shapes	9,757,418
9	ptv	stop_times	8,122,810
10	ptv	stops	27,821
11	ptv	trips	236,613

2 Data Preprocessing

2.1

```
create table ptv.mb2021_mel
as
select * from mb_2021 m2
where gcc_name21 = 'Greater Melbourne';
```

2.2

```
create table lga_2021_mel_geom as
SELECT l.*,m.wkb_geometry
FROM ptv.mb2021_mel m
join ptv.lga_2021 l
on m.mb_code21 = l.mb_code_2021;
```

```sql
create table sal_2021_mel_geom as
SELECT s.*,m.wkb_geometry
FROM ptv.mb2021_mel m
join ptv.sal_2021 s
on m.mb_code21 = s.mb_code_2021;
```

2.3

```sql
create table Melbourne_Metropolitan as
SELECT ST_Union(wkb_geometry) AS melbourne_metropolitan_boundary
FROM ptv.mb2021_mel;

select * from Melbourne_Metropolitan;
```

2.4

```sql
SELECT AddGeometryColumn('stops', 'geom', 7844, 'POINT', 2);

UPDATE ptv.stops
SET geom = ST_SetSRID(ST_MakePoint(stop_lon, stop_lat), 7844);

select * from ptv.stops;
```

2.5

```sql
Create table stops_routes as
select s.*, r.route_short_name, r.route_long_name,
CASE WHEN r.route_type = '0' THEN 'tram'
WHEN r.route_type = '2' THEN 'train'
WHEN r.route_type = '3' THEN 'bus'
ELSE 'Unknown' END AS vehicle_type
from stops s
join stop_times st
on s.stop_id = st.stop_id
join trips t
on st.trip_id = t.trip_id
join routes r
on t.route_id = r.route_id;
```

3 Data Analysis and Visualization

3.1

```
create table lga_gap as
select * from lga_geom sg
where lga_name_2021 not in (select distinct lga_name_2021 from lga_stop);
```

```
create table sal_gap as
select * from sal_geom sg
where sal_name_2021 not in (select distinct sal_name_2021 from sal_stop);
```

3.2

```
create table lga_geom as
select lga_name_2021 , ST_Union(wkb_geometry) AS melbourne_metropolitan_boundary from ptv.lga_2021_mel_geom lmg
group by lga_name_2021;

select * from lga_geom;
```

```
create table lga_stop as
select lga_name_2021,melbourne_metropolitan_boundary ,
count(distinct stop_id)
from
ptv.lga_geom,
ptv.stops
where st_within(geom, melbourne_metropolitan_boundary)
group by lga_name_2021, melbourne_metropolitan_boundary;

select * from ptv.lga_stop;
```

```
create table sal_geom as
select sal_name_2021 , ST_Union(wkb_geometry) AS melbourne_metropolitan_boundary from ptv.sal_2021_mel_geom
group by sal_name_2021;

select * from sal_geom;
```

```sql
create table sal_stop as
select sal_name_2021,melbourne_metropolitan_boundary ,
count(distinct stop_id) as stop_counts
from
ptv.sal_geom,
ptv.stops
where st_within(geom, melbourne_metropolitan_boundary)
group by sal_name_2021, melbourne_metropolitan_boundary;

select * from ptv.sal_stop;
```

3.3

```sql
drop table stop_vehicle ;
create table stop_vehicle as
select distinct stop_id, geom,
case when vehicle_type = 'bus' then 1 else 0 end as bus,
case when vehicle_type = 'train' then 1 else 0 end as train,
case when vehicle_type = 'tram' then 1 else 0 end as tram
from stops_routes sr;
```

```sql
drop table lga_vehicle_stop ;
create table lga_vehicle_stop as
select t1.*, t2.area_sqkm from
(select lga_name_2021,
melbourne_metropolitan_boundary,
sum(bus) as bus,
sum(train) as train,
sum(tram) as tram
from
ptv.lga_geom,
ptv.stop_vehicle
where st_within(geom, melbourne_metropolitan_boundary)
group by lga_name_2021, melbourne_metropolitan_boundary)t1
join
(select lga_name_2021, SUM(CAST(area_albers_sqkm AS FLOAT)) AS area_sqkm from lga_2021_mel_geom lmg
group by lga_name_2021)t2
on t1.lga_name_2021 = t2.lga_name_2021;
```

```sql
drop table lga_bus;
create table lga_bus as
select * from
(select lga_name_2021,melbourne_metropolitan_boundary,bus, area_sqkm, bus/area_sqkm as bus_ratio from lga_vehicle_stop
order by bus/area_sqkm desc
limit 5) t1;
select lga_name_2021, bus, area_sqkm, bus_ratio from lga_bus;
```

_bus 1

ct lga_name_2021, bus, area_sqkm, bu | Enter a SQL expression to filter results (use Ctrl+Space)

lga_name_2021	bus	area_sqkm	bus_ratio
Moreland	704	50.9522	13.8168714992
Darebin	720	53.4712	13.4651924774
Glen Eira	517	38.6908	13.3623497059
Whitehorse	842	64.2801	13.0989217503
Maribyrnong	396	31.2258	12.681820802

```sql
create table lga_train as
select * from
(select lga_name_2021,melbourne_metropolitan_boundary,train, area_sqkm, train/area_sqkm as train_ratio from lga_vehicle_st
order by train/area_sqkm desc
limit 5) t1;

select lga_name_2021, train, area_sqkm, train_ratio from lga_train;
```

_train 1

ect lga_name_2021, train, area_sqkm, tr | Enter a SQL expression to filter results (use Ctrl+Space)

lga_name_2021	train	area_sqkm	train_ratio
Stonnington	15	25.651	0.5847725235
Yarra	10	19.5421	0.5117157317
Melbourne	16	37.5452	0.4261530102
Glen Eira	11	38.6908	0.2843053129
Moreland	13	50.9522	0.255141093

```sql
create table lga_tram as
select * from
(select lga_name_2021,melbourne_metropolitan_boundary,tram, area_sqkm, tram/area_sqkm as tram_ratio from lga_vehicle_stop
order by tram/area_sqkm desc
limit 5) t1;


select lga_name_2021, tram, area_sqkm, tram_ratio from lga_tram;
```

_tram 1

lect lga_name_2021, tram, area_sqkm, tr | Enter a SQL expression to filter results (use Ctrl+Space)

lga_name_2021	tram	area_sqkm	tram_ratio
Stonnington	221	25.651	8.6156485127
Port Phillip	164	20.6141	7.9557196288
Yarra	151	19.5421	7.7269075483
Melbourne	286	37.5452	7.617485058
Boroondara	243	60.1776	4.0380473798

```sql
create table sal_vehicle_stop as
select t1.*, t2.area_sqkm from
(select sal_name_2021,
melbourne_metropolitan_boundary,
sum(bus) as bus,
sum(train) as train,
sum(tram) as tram
from
ptv.sal_geom,
ptv.stop_vehicle
where st_within(geom, melbourne_metropolitan_boundary)
group by sal_name_2021, melbourne_metropolitan_boundary)t1
join
(select sal_name_2021, SUM(CAST(area_albers_sqkm AS FLOAT)) AS area_sqkm from sal_2021_mel_geom lmg
group by sal_name_2021)t2
on t1.sal_name_2021 = t2.sal_name_2021;
```

```sql
create table sal_bus as
select * from
(select sal_name_2021,melbourne_metropolitan_boundary,bus, area_sqkm, bus/area_sqkm as bus_ratio from sal_vehicle_stop
order by bus/area_sqkm desc
limit 5) t1;


select sal_name_2021, bus, area_sqkm, bus_ratio from sal_bus;
```

al_bus 1 ×

elect sal_name_2021, bus, area_sqkm, bu: Enter a SQL expression to filter results (use Ctrl+Space)

	sal_name_2021	bus	area_sqkm	bus_ratio
1	Gardenvale	9	0.2588	34.7758887172
2	Ripponlea	8	0.2858	27.9916025192
3	Brunswick West	83	3.2174	25.7972275751
4	South Kingsville	16	0.6368	25.1256281407
5	Box Hill (Vic.)	88	3.522	24.9858035207

```sql
create table sal_train as
select * from
(select sal_name_2021,melbourne_metropolitan_boundary,train, area_sqkm, train/area_sqkm as train_ratio from sal_vehicle_st
order by train/area_sqkm desc
limit 5) t1;

select sal_name_2021, train, area_sqkm, train_ratio from sal_train;
```

_train 1 ×

ect sal_name_2021, train, area_sqkm, tra Enter a SQL expression to filter results (use Ctrl+Space)

	sal_name_2021	train	area_sqkm	train_ratio
	Ripponlea	1	0.2858	3.4989503149
	Kooyong	1	0.5061	1.9758940921
	Caulfield East	2	1.2587	1.5889409708
	Cremorne (Vic.)	1	0.6799	1.4708045301
	Balaclava (Vic.)	1	0.7638	1.3092432574

```sql
create table sal_tram as
select * from
(select sal_name_2021,melbourne_metropolitan_boundary,tram, area_sqkm, tram/area_sqkm as tram_ratio from sal_vehicle_stop
order by tram/area_sqkm desc
limit 5) t1;


select sal_name_2021, tram, area_sqkm, tram_ratio from sal_tram;
```

tram 1 ×

ect sal_name_2021, tram, area_sqkm, tr… Enter a SQL expression to filter results (use Ctrl+Space)

sal_name_2021	tram	area_sqkm	tram_ratio
Fitzroy (Vic.)	28	1.4071	19.8990832208
Travancore	8	0.4184	19.120458891
Melbourne	124	6.5866	18.8261014788
St Kilda (Vic.)	58	3.1785	18.2476010697
Southbank	26	1.564	16.6240409207

3.4

```sql
create table melbourne_lga_mesh as
select * from lga_2021_mel_geom
where upper(lga_name_2021) in ('MORELAND','MOONEE VALLEY','MELBOURNE','MARIBYRNONG',
'HOBSONS BAY','STONNINGTON','PORT PHILLIP');
```

```sql
create table melbourne_lga_stops as
select distinct stop_id, geom, vehicle_type from stops_routes_vehicle,lga_melbourne_geom
where st_within(geom, melbourne_metropolitan_boundary);
```

```sql
create table lga_melbourne_geom as
select * from lga_geom
where upper(lga_name_2021) in ('MORELAND','MOONEE VALLEY','MELBOURNE','MARIBYRNONG',
'HOBSONS BAY','STONNINGTON','PORT PHILLIP');
```