POC: OSM ingestion to postGIS database

1. PostGIS setup

```
# 1. Start and enable PostgreSQL service
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Install the PostGIS extension
sudo apt-get update
sudo apt-get install postgis postgresql-14-postgis-3
sudo systemctl restart postgresql

# 2. Access the PostgreSQL prompt as the default postgres user
sudo -u postgres psql

# 3. Inside the PostgreSQL prompt, run the following SQL commands:

# Create a new database
CREATE DATABASE osm_database;

# Create a new user with a password
CREATE USER yzpt WITH PASSWORD 'pwd';

# Grant the user privileges on the database
GRANT ALL PRIVILEGES ON DATABASE osm_database TO yzpt;

# Connect to the new database
\c osm_database;

# Enable the PostGIS extension
CREATE EXTENSION postgis;

# Exit PostgreSQL prompt
\q
```

2. Python process

2.1. import modules

In [5]:

```
import pandas as pd
import geopandas as pd
import osmnx as ox
import geopandas as gpd
from sqlalchemy import create_engine
```

2.2. Create connection to the database

In [6]:

```
# Define town name and database connection parameters
town_name = "Leforest, France"  # Specify the desired town
db_name = "osm_database"
db_user = "yzpt"
db_password = "pwd"
db_host = "localhost"  # Or the host IP if it's remote
db_port = "5432"       # Default PostgreSQL port

# Set up the database connection using SQLAlchemy
engine = create_engine(f"postgresql+psycopg2://{db_user}:{db_password}@{db_host}:{db_port}/{db_name}")
```

2.3. OSM data download for the town (Leforest, France, in this case) and postGIS ingestion

In [3]:

```
# Download the street network data for the specified town
G = ox.graph_from_place(town_name, network_type="all")

# Convert the network graph to a GeoDataFrame (edges only)
edges = ox.graph_to_gdfs(G, nodes=False)

# Save the GeoDataFrame to a file
# edges.to_file("edges.shp")
# Save the street layer (edges) into the PostgreSQL database as a table named "osm_streets"
edges.to_postgis("osm_streets_leforest", con=engine, if_exists="replace", index=False)
print("OSM streets data successfully saved to PostgreSQL!")
```

2.4. Query the OSM data from the database

In [7]:

```
query = "SELECT * FROM osm_streets_leforest;"
gdf = gpd.read_postgis(query, engine, geom_col='geometry')
```

In [15]:

```
gdf.head()
```

Out[15]:

	osmid	name	highway	oneway	reversed	length	geometry	junction	ref	maxspeed	access
0	107901967	Rue de Dury	residential	False	True	202.152	LINESTRING (3.05885 50.44392, 3.06047 50.44344...	None	None	None	None
1	107916385	Rue d'Amiens	residential	False	True	53.273	LINESTRING (3.05885 50.44392, 3.05882 50.44388...	None	None	None	None
2	1009079423	Rue d'Amiens	residential	False	False	64.675	LINESTRING (3.05885 50.44392, 3.05894 50.44404...	None	None	None	None
3	107901971	Résidence Domaine du Bois	residential	False	True	95.189	LINESTRING (3.06241 50.44377, 3.06345 50.44431)	None	None	None	None
4	107901973	None	residential	True	False	8.168	LINESTRING (3.05873 50.44114, 3.05874 50.44116...	roundabout	None	None	None

2.5. Quick visualization of the street network

In [18]:

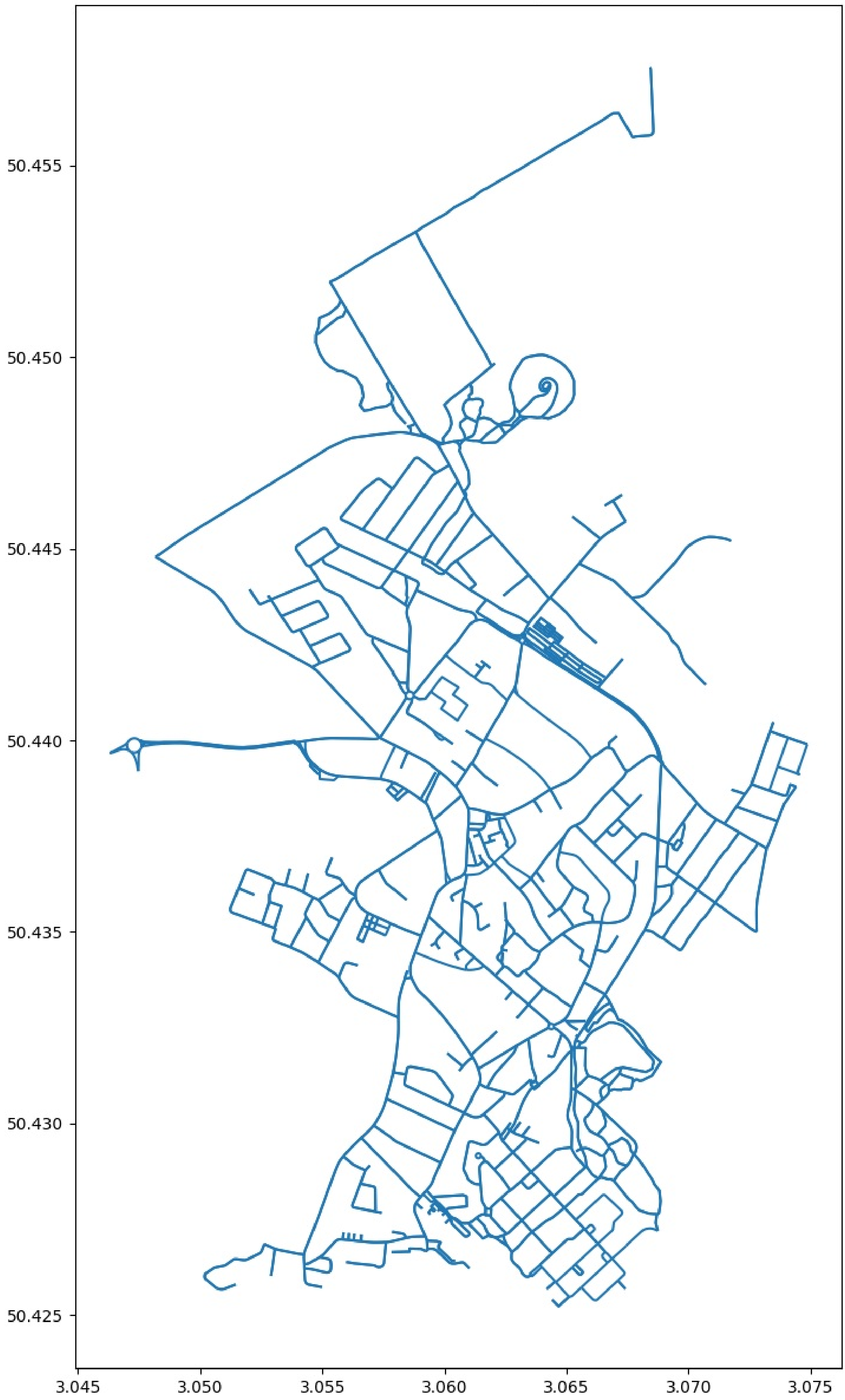
```
import matplotlib.pyplot as plt

# Plot the GeoDataFrame
ax = gdf.plot(figsize=(15, 15))

# Set the title
ax.set_title("OSM streets in Leforest, France")

# Show the plot
plt.show()
```

OSM streets in Leforest, France


50.455
50.450
50.445
50.440
50.435
50.430
50.425
3.045
3.050
3.055
3.060
3.065
3.070
3.075