KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPATMENT OF GEOMATICS ENGINEERING

Final Report on
Determining Shortest Route
Using pgRouting, PostGIS, PostgreSQL and QGIS

Submitted By

Rajan Paudel
Roll no: 37
3rd year,1st semester

Submitted To

Mr. Ajay Thapa
Department of Geomatics Engineering

Date of Submission: 2024/06/14

Abstract

This project focuses on determining the shortest route using the integrated capabilities of pgRouting, PostGIS, PostgreSQL, and QGIS. By extending PostgreSQL with PostGIS, we enable support for geographic objects, while pgRouting provides advanced routing algorithms. The process involves setting up a spatially-enabled database, importing and processing road network data, creating routing functions, and visualizing the results in QGIS. This integration offers a robust solution for spatial analysis and route optimization, demonstrating its potential applications in urban planning, transportation, and logistics. The project highlights the efficiency and practicality of using open-source GIS tools for complex spatial queries.

Table Of Content

1. Introduction

PostGIS is an extension for PostgreSQL that adds support for geographic objects and spatial analysis. It's commonly used for storing and manipulating geographical data within a PostgreSQL database. pgRouting is an extension to PostGIS that provides routing functionality, allowing you to calculate shortest paths, among other routing tasks. QGIS, on the other hand, is a popular open-source Geographic Information System (GIS) software that can be used for visualizing and analyzing spatial data.

Finding the shortest route between two points means figuring out the quickest or easiest way to get from one place to another. This is important for things like GPS navigation, delivery planning, and designing networks. To find this route, we use special methods called algorithms.

Dijkstra's algorithm is a method for finding the shortest path between two points on a map or network. It starts at the origin point, marking it with a distance of zero. From there, it examines all neighboring points, choosing the one with the smallest distance. It updates the shortest known distances to each neighboring point and marks the current point as visited. This process repeats, always selecting the closest unvisited point and updating distances, until the destination is reached or all points have been visited. By systematically exploring and updating distances, Dijkstra's algorithm ensures that the shortest path is found efficiently.

To find the best route, we need to know where we’re starting, where we want to go, and any special preferences, like finding the fastest route. By setting these details, we can easily find the best path to our destination.

PostgreSQL

PostgreSQL is a powerful, free database system that helps store and manage data efficiently. It's known for being reliable and flexible, making it great for a variety of uses, from running websites to handling large datasets and mapping information. It ensures data accuracy and can handle many tasks at once without slowing down. With PostgreSQL, you can create custom functions and use different programming languages. It also supports copying data across servers for backup and scaling. Installing it is easy on different operating systems like Windows, Mac, and Linux.

pgRouting

pgRouting is an extension for PostgreSQL and PostGIS that adds routing and network analysis features to your spatial databases. This means you can calculate the best routes and analyze networks of roads or paths. It's useful for things like planning transportation routes, optimizing delivery paths, and analyzing traffic. This makes it a powerful tool for anyone working with maps and routes.

QGIS

QGIS is a free and open-source Geographic Information System (GIS) software that helps you create, edit, visualize, and analyze geographic data. It's used for making maps and performing spatial analysis, making it great for projects in urban planning, environmental management, and transportation. With QGIS, you can work with different types of geographic data, such as satellite images and vector files like shapefiles. It supports a wide range of file formats and has many plugins to add extra features.

osm2po

osm2po is a tool that converts OpenStreetMap (OSM) data into a format that can be used for routing and navigation. It processes the raw map data from OSM and creates a database that you can use to find the best routes for driving, walking, or biking. osm2po is useful for projects that need detailed map data and routing capabilities, like creating GPS applications or mapping services. It works with PostgreSQL and PostGIS to store and analyze the data. This makes it easier to integrate OSM data into your applications and perform advanced routing and navigation tasks.

Java

Java is a widely-used, high-level programming language known for its portability, reliability, and performance. OpenStreetMap (OSM) provides open-source geographic data that you can use in Java applications to create maps, perform geospatial analysis, and build location-based services.

2. Objectives

2.1 Primary Objectives

- Calculate the shortest route from Baneshwor to Kathmandu University.

2.2 Secondary Objectives

The secondary objectives are:

- To use OSM, QGIS, PostgreSQL, and PostGIS to solve real-world problems.
- To deepen the understanding of spatial database and their functionalities.
- To visualize spatial data and develop practical GIS skills.

3. Methodology

3.1 Study area

The study area encompasses the route between Baneshwor and Kathmandu University, Kavrepalanchok located in Bagmati province of Nepal. The specific coordinates for the starting point is (27.703396, 85.339894) and the ending point (Kathmandu University) is (27.620602,85.538359).

Figure 1 : Study Area

3.2 Materials Used

3.2.1 Data used

OSM data for Nepal, which includes detailed information about the road network and other relevant geographic features was downloaded online.

3.2.2 Software used

The softwares used for this project are:

- PostgreSQL: A powerful relational database management system used for storing and managing spatial data.
- PostGIS: An extension of PostgreSQL that adds support for geographic objects, enabling spatial queries and analysis.
- pgRouting: An extension of PostGIS that provides routing and network analysis capabilities.
- QGIS: An open-source geographic information system used for data visualization, spatial analysis, and map creation.

3.3 Work Flow

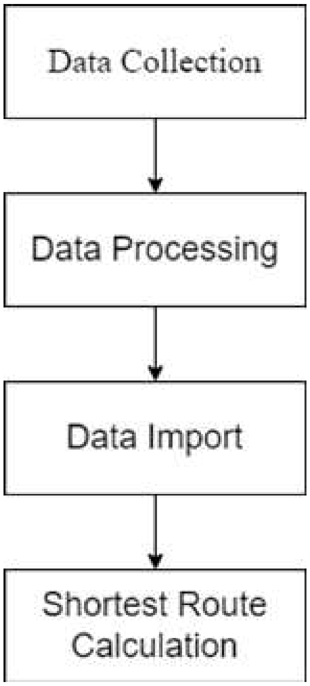
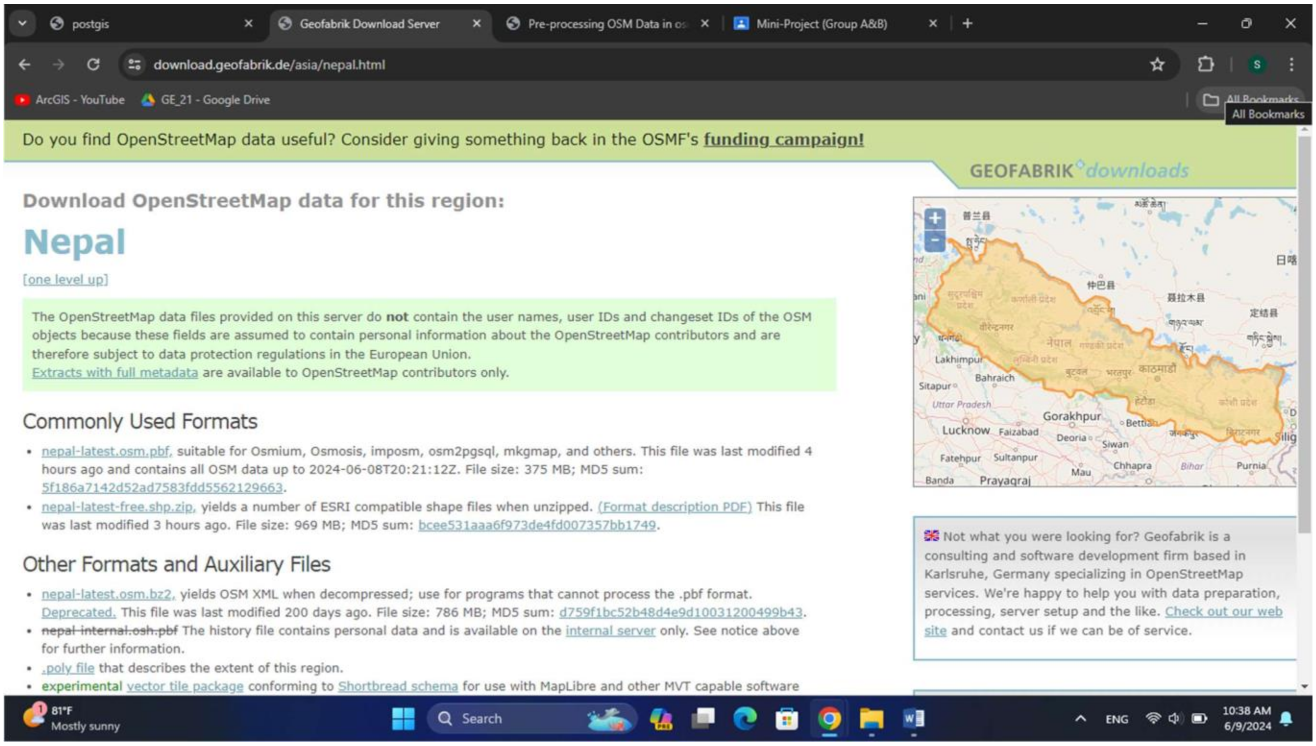


Figure 2 : Work Flow

The step-by-step explanation of each process is as follows:

3.3.1 Data Collection

OSM data for Nepal was downloaded through the link https://download.geofabrik.de/asia.html in pbf format.



Do you find OpenStreetMap data useful? Consider giving something back in the OSMF's **funding campaign!**

Download OpenStreetMap data for this region:

Nepal

[one level up]

The OpenStreetMap data files provided on this server do **not** contain the user names, user IDs and changeset IDs of the OSM objects because these fields are assumed to contain personal information about the OpenStreetMap contributors and are therefore subject to data protection regulations in the European Union.
Extracts with full metadata are available to OpenStreetMap contributors only.

Commonly Used Formats

- nepal-latest.osm.pbf, suitable for Osmium, Osmosis, imposm, osm2pgsql, mkgmap, and others. This file was last modified 4 hours ago and contains all OSM data up to 2024-06-08T20:21:12Z. File size: 375 MB; MD5 sum: 5f186a7142d52ad7583fdd5562129663.
- nepal-latest-free.shp.zip, yields a number of ESRI compatible shape files when unzipped. (Format description PDF) This file was last modified 3 hours ago. File size: 969 MB; MD5 sum: bcee531aaa6f973de4fd007357bb1749.

Other Formats and Auxiliary Files

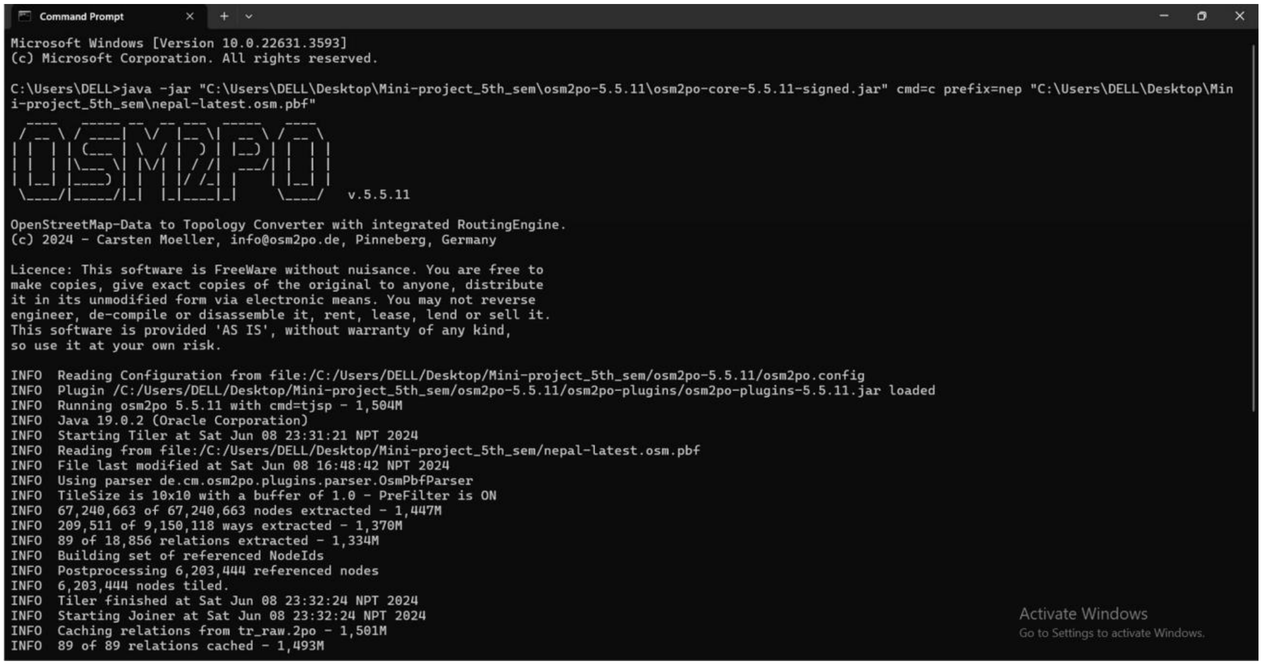
- nepal-latest.osm.bz2, yields OSM XML when decompressed; use for programs that cannot process the .pbf format. Deprecated. This file was last modified 200 days ago. File size: 786 MB; MD5 sum: d759f1bc52b48d4e9d10031200499b43.
- ~~nepal-internal.osh.pbf~~ The history file contains personal data and is available on the internal server only. See notice above for further information.
- .poly file that describes the extent of this region.
- experimental vector tile package conforming to Shortbread schema for use with MapLibre and other MVT capable software

Not what you were looking for? Geofabrik is a consulting and software development firm based in Karlsruhe, Germany specializing in OpenStreetMap services. We're happy to help you with data preparation, processing, server setup and the like. Check out our web site and contact us if we can be of service.

Figure 3 : Download Data From OSM

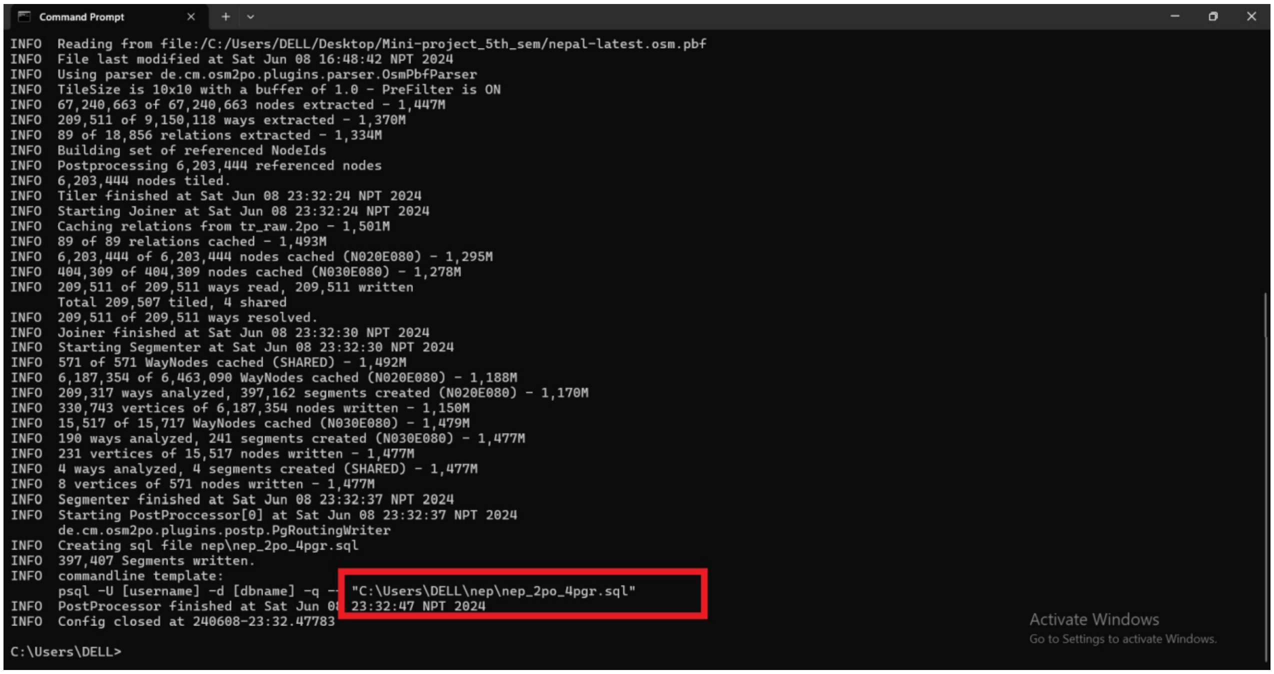
3.3.2 Data Preprocessing

The OSM data of Nepal was converted to usable form using command prompt and osm2po. osm2po is a powerful tool designed to convert OpenStreetMap (OSM) data into a format that can be used for routing and network analysis in PostgreSQL with PostGIS and pgRouting.



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>java -jar "C:\Users\DELL\Desktop\Mini-project_5th_sem\osm2po-5.5.11\osm2po-core-5.5.11-signed.jar" cmd=c prefix=nep "C:\Users\DELL\Desktop\Min
i-project_5th_sem\nepal-latest.osm.pbf"
  ____   _____ __  __ ___  _____   ____
 / __ \ / ____|  \/  |__ \|  __ \ / __ \
| |  | | (___ | \  / |  ) | |__) | |  | |
| |  | |\___ \| |\/| | / /|  ___/| |  | |
| |__| |____) | |  | |/ /_| |    | |__| |
 \____/|_____/|_|  |_|____|_|     \____/   v.5.5.11

OpenStreetMap-Data to Topology Converter with integrated RoutingEngine.
(c) 2024 - Carsten Moeller, info@osm2po.de, Pinneberg, Germany

Licence: This software is FreeWare without nuisance. You are free to
make copies, give exact copies of the original to anyone, distribute
it in its unmodified form via electronic means. You may not reverse
engineer, de-compile or disassemble it, rent, lease, lend or sell it.
This software is provided 'AS IS', without warranty of any kind,
so use it at your own risk.

INFO  Reading Configuration from file:/C:/Users/DELL/Desktop/Mini-project_5th_sem/osm2po-5.5.11/osm2po.config
INFO  Plugin /C:/Users/DELL/Desktop/Mini-project_5th_sem/osm2po-5.5.11/osm2po-plugins/osm2po-plugins-5.5.11.jar loaded
INFO  Running osm2po 5.5.11 with cmd=tjsp - 1,504M
INFO  Java 19.0.2 (Oracle Corporation)
INFO  Starting Tiler at Sat Jun 08 23:31:21 NPT 2024
INFO  Reading from file:/C:/Users/DELL/Desktop/Mini-project_5th_sem/nepal-latest.osm.pbf
INFO  File last modified at Sat Jun 08 16:48:42 NPT 2024
INFO  Using parser de.cm.osm2po.plugins.parser.OsmPbfParser
INFO  TileSize is 10x10 with a buffer of 1.0 - PreFilter is ON
INFO  67,240,663 of 67,240,663 nodes extracted - 1,447M
INFO  209,511 of 9,150,118 ways extracted - 1,370M
INFO  89 of 18,856 relations extracted - 1,334M
INFO  Building set of referenced NodeIds
INFO  Postprocessing 6,203,444 referenced nodes
INFO  6,203,444 nodes tiled.
INFO  Tiler finished at Sat Jun 08 23:32:24 NPT 2024
INFO  Starting Joiner at Sat Jun 08 23:32:24 NPT 2024
INFO  Caching relations from tr_raw.2po - 1,501M
INFO  89 of 89 relations cached - 1,493M
```

Figure 4 : Preprocessing OSM data



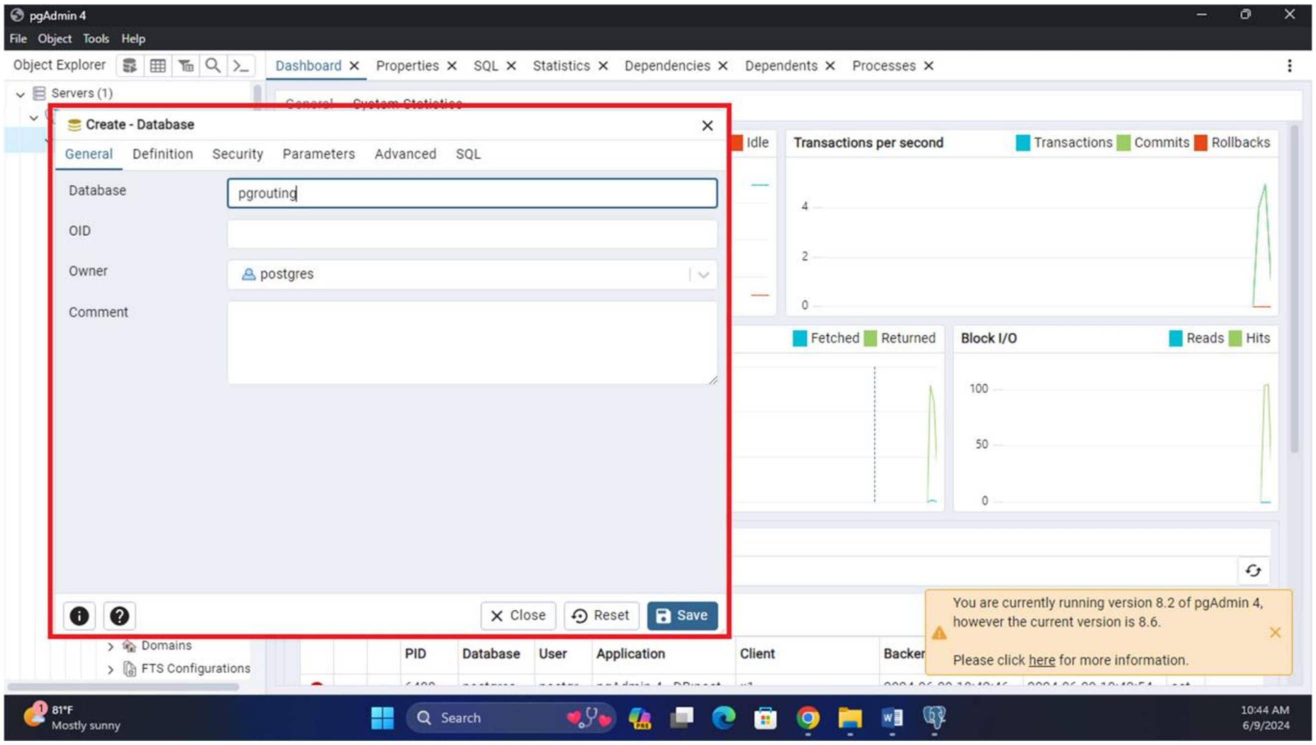
```
INFO  Reading from file:/C:/Users/DELL/Desktop/Mini-project_5th_sem/nepal-latest.osm.pbf
INFO  File last modified at Sat Jun 08 16:48:42 NPT 2024
INFO  Using parser de.cm.osm2po.plugins.parser.OsmPbfParser
INFO  TileSize is 10x10 with a buffer of 1.0 - PreFilter is ON
INFO  67,240,663 of 67,240,663 nodes extracted - 1,447M
INFO  209,511 of 9,150,118 ways extracted - 1,370M
INFO  89 of 18,856 relations extracted - 1,334M
INFO  Building set of referenced NodeIds
INFO  Postprocessing 6,203,444 referenced nodes
INFO  6,203,444 nodes tiled.
INFO  Tiler finished at Sat Jun 08 23:32:24 NPT 2024
INFO  Starting Joiner at Sat Jun 08 23:32:24 NPT 2024
INFO  Caching relations from tr_raw.2po - 1,501M
INFO  89 of 89 relations cached - 1,493M
INFO  6,203,444 of 6,203,444 nodes cached (N020E080) - 1,295M
INFO  404,309 of 404,309 nodes cached (N030E080) - 1,278M
INFO  209,511 of 209,511 ways read, 209,511 written
      Total 209,507 tiled, 4 shared
INFO  209,511 of 209,511 ways resolved.
INFO  Joiner finished at Sat Jun 08 23:32:30 NPT 2024
INFO  Starting Segmenter at Sat Jun 08 23:32:30 NPT 2024
INFO  571 of 571 WayNodes cached (SHARED) - 1,492M
INFO  6,187,354 of 6,463,090 WayNodes cached (N020E080) - 1,188M
INFO  209,317 ways analyzed, 397,162 segments created (N020E080) - 1,170M
INFO  330,743 vertices of 6,187,354 nodes written - 1,150M
INFO  15,517 of 15,717 WayNodes cached (N030E080) - 1,479M
INFO  190 ways analyzed, 241 segments created (N030E080) - 1,477M
INFO  231 vertices of 15,517 nodes written - 1,477M
INFO  4 ways analyzed, 4 segments created (SHARED) - 1,477M
INFO  8 vertices of 571 nodes written - 1,477M
INFO  Segmenter finished at Sat Jun 08 23:32:37 NPT 2024
INFO  Starting PostProccessor[0] at Sat Jun 08 23:32:37 NPT 2024
      de.cm.osm2po.plugins.postp.PgRoutingWriter
INFO  Creating sql file nep\nep_2po_4pgr.sql
INFO  397,407 Segments written.
INFO  commandline template:
      psql -U [username] -d [dbname] -q -f "C:\Users\DELL\nep\nep_2po_4pgr.sql"
INFO  PostProcessor finished at Sat Jun 08 23:32:47 NPT 2024
INFO  Config closed at 240608-23:32.47783

C:\Users\DELL>
```

Figure 5 : Navigating converted data

3.3.3 Data Import

To import data from OSM to PostgreSQL, first of all a new database named “pgrouting” was created. Then two extensions “postgis” and “pgrouting” was created.

Then, the SQL file of converted OSM data was run in pgadmin to create and insert data into table.



Create - Database

General Definition Security Parameters Advanced SQL

Database: pgrouting

OID:

Owner: postgres

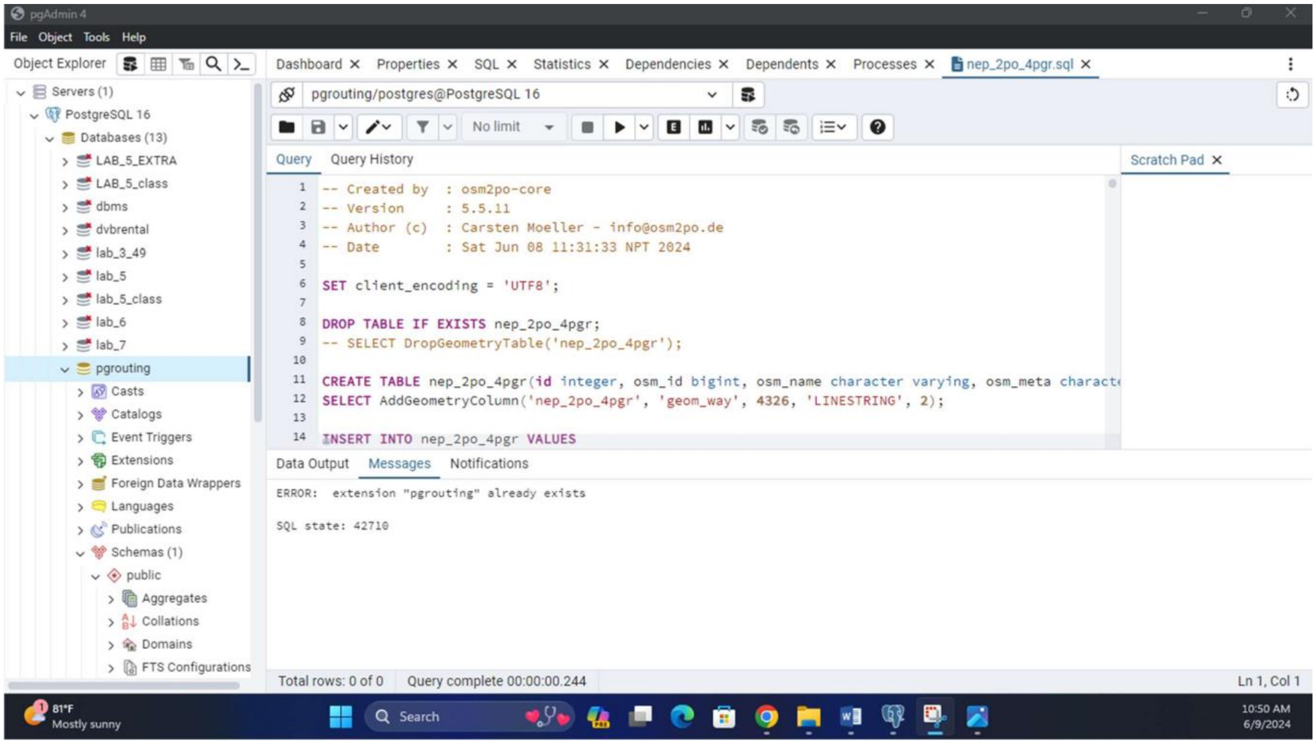
Comment:

Close Reset Save

You are currently running version 8.2 of pgAdmin 4, however the current version is 8.6.

Please click here for more information.

Figure 6 : Creating database "pgrouting”



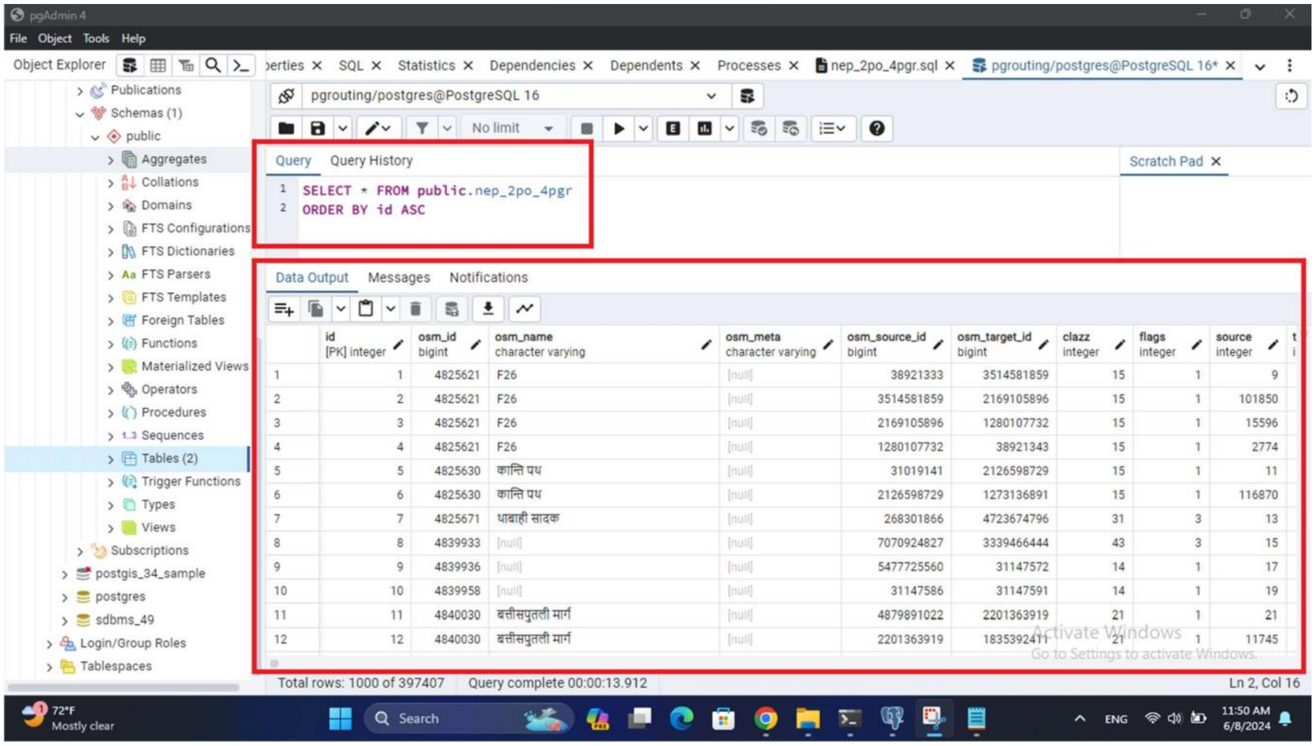
pgrouting/postgres@PostgreSQL 16

```
-- Created by  : osm2po-core
-- Version     : 5.5.11
-- Author (c)  : Carsten Moeller - info@osm2po.de
-- Date        : Sat Jun 08 11:31:33 NPT 2024

SET client_encoding = 'UTF8';

DROP TABLE IF EXISTS nep_2po_4pgr;
-- SELECT DropGeometryTable('nep_2po_4pgr');

CREATE TABLE nep_2po_4pgr(id integer, osm_id bigint, osm_name character varying, osm_meta characte
SELECT AddGeometryColumn('nep_2po_4pgr', 'geom_way', 4326, 'LINESTRING', 2);

INSERT INTO nep_2po_4pgr VALUES
```

```
ERROR:  extension "pgrouting" already exists

SQL state: 42710
```

Total rows: 0 of 0   Query complete 00:00:00.244   Ln 1, Col 1

Figure 7 : Open data into table

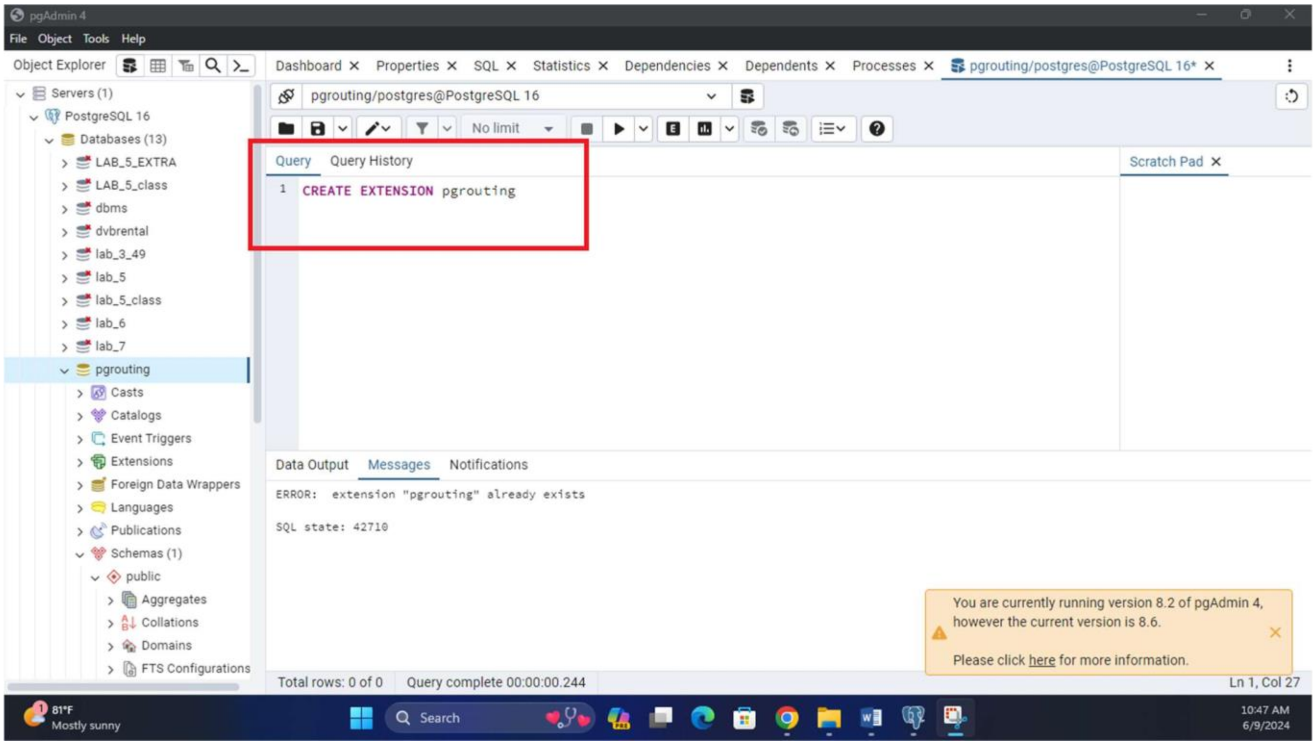


```
SELECT * FROM public.nep_2po_4pgr
ORDER BY id ASC
```

	id [PK] integer	osm_id bigint	osm_name character varying	osm_meta character varying	osm_source_id bigint	osm_target_id bigint	clazz integer	flags integer	source integer
1	1	4825621	F26	[null]	38921333	3514581859	15	1	9
2	2	4825621	F26	[null]	3514581859	2169105896	15	1	101850
3	3	4825621	F26	[null]	2169105896	1280107732	15	1	15596
4	4	4825621	F26	[null]	1280107732	38921343	15	1	2774
5	5	4825630	कान्ति पथ	[null]	31019141	2126598729	15	1	11
6	6	4825630	कान्ति पथ	[null]	2126598729	1273136891	15	1	116870
7	7	4825671	थावाही सादक	[null]	268301866	4723674796	31	3	13
8	8	4839933	[null]	[null]	7070924827	3339466444	43	3	15
9	9	4839936	[null]	[null]	5477725560	31147572	14	1	17
10	10	4839958	[null]	[null]	31147586	31147591	14	1	19
11	11	4840030	बत्तीसपुतली मार्ग	[null]	4879891022	2201363919	21	1	21
12	12	4840030	बत्तीसपुतली मार्ग	[null]	2201363919	1835392411	21	1	11745

Total rows: 1000 of 397407 Query complete 00:00:13.912

Figure 8 : Simple Query and Output



```
CREATE EXTENSION pgrouting
```

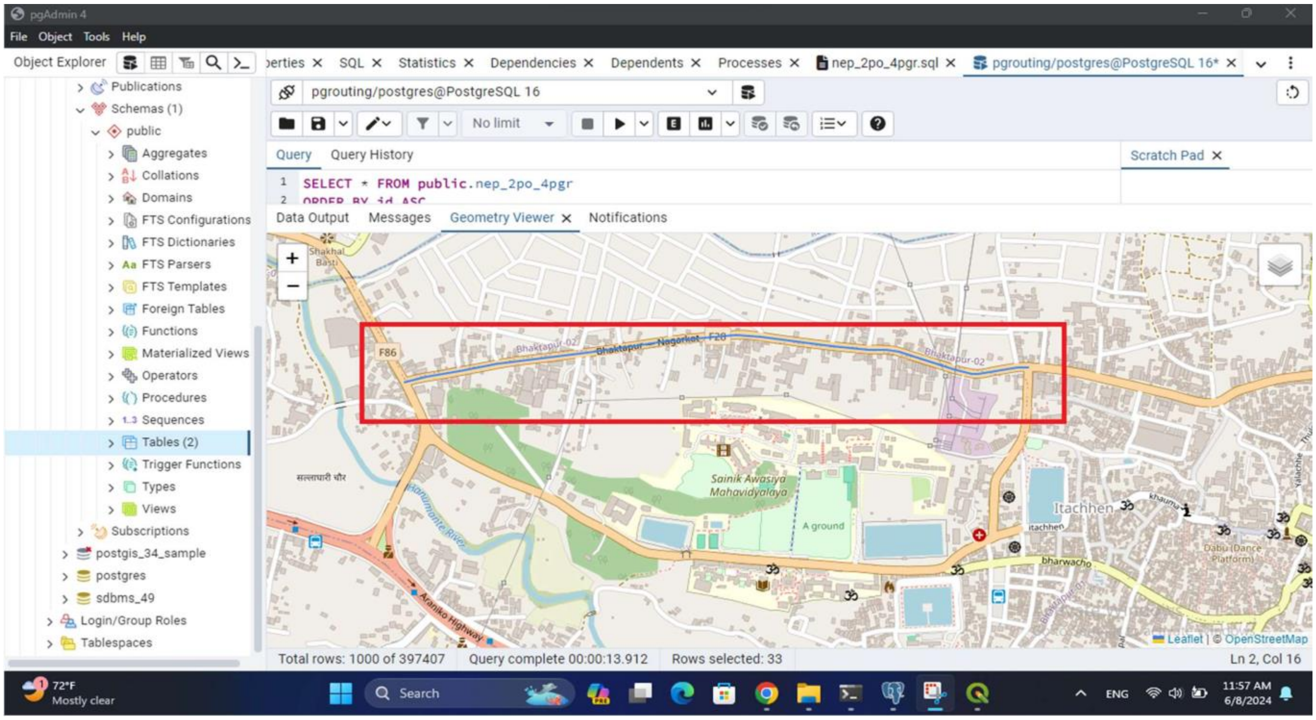
```
ERROR:  extension "pgrouting" already exists

SQL state: 42710
```

You are currently running version 8.2 of pgAdmin 4, however the current version is 8.6.

Please click here for more information.

Total rows: 0 of 0 Query complete 00:00:00.244

Figure 9 : Create Extension pgrouting



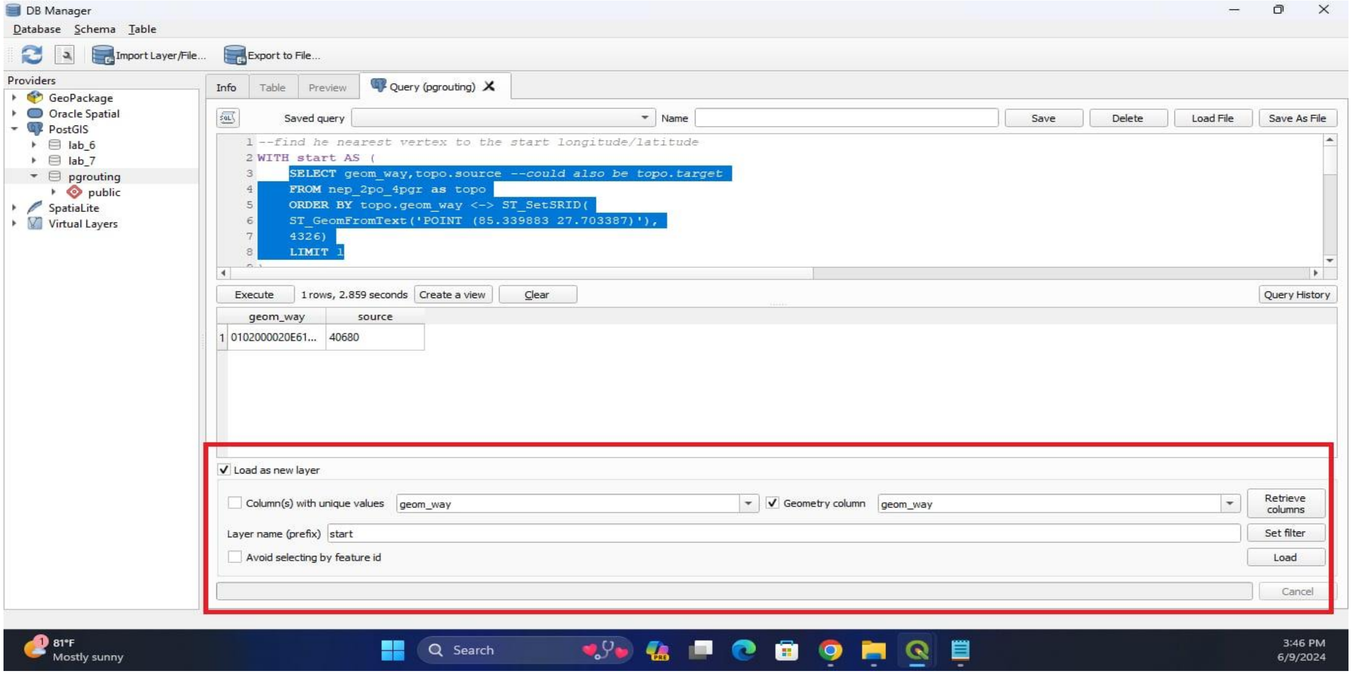
```
SELECT * FROM public.nep_2po_4pgr
ORDER BY id ASC
```

Total rows: 1000 of 397407   Query complete 00:00:13.912   Rows selected: 33   Ln 2, Col 16

Figure 10 : Simple Query and Output on Geometry

3.3.4 Route Calculation

For this, I have Selected my house location (27.676378, 85.385425) as the Origin and Kathmandu University (27.620602,85.538359) as the destination. First of all, database was connected with PostgreSQL using a username and password. The OSM data of routes was loaded in QGIS. After that base map was added. In QGIS DB manager, query was executed to determine the shortest route using pgrouting and three layers was added to QGIS.



```
--find he nearest vertex to the start longitude/latitude
WITH start AS (
    SELECT geom_way,topo.source --could also be topo.target
    FROM nep_2po_4pgr as topo
    ORDER BY topo.geom_way <-> ST_SetSRID(
    ST_GeomFromText('POINT (85.339883 27.703387)'),
    4326)
    LIMIT 1
```

1 rows, 2.859 seconds

	geom_way	source
1	0102000020E61...	40680

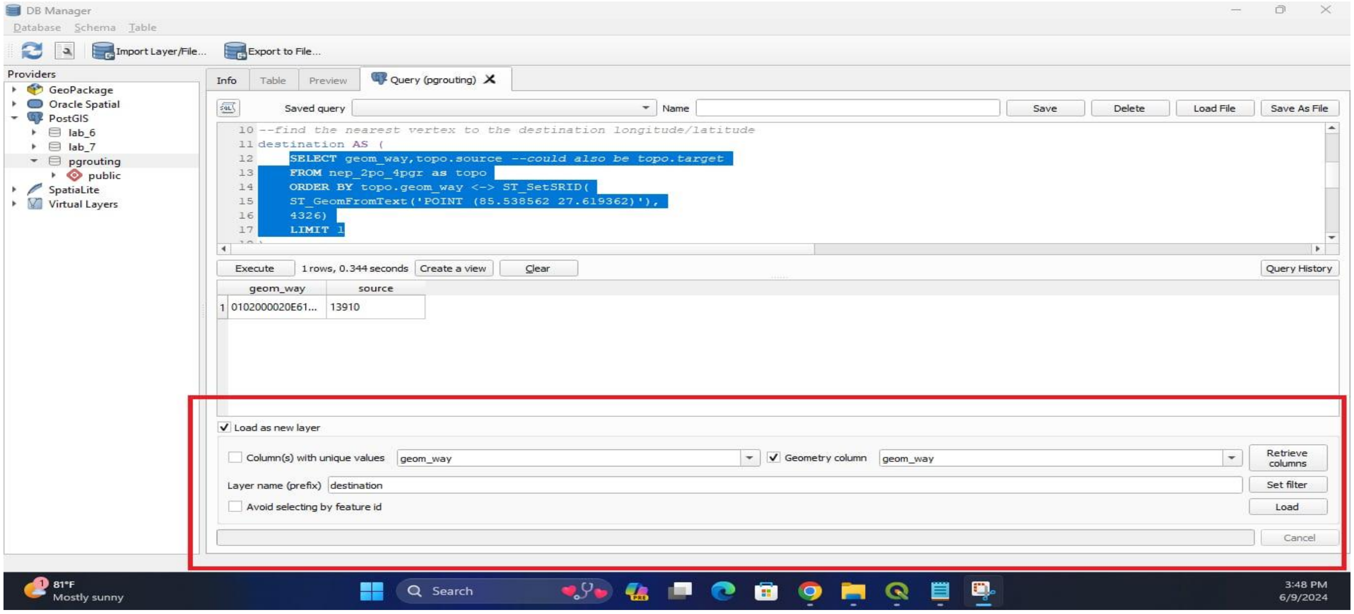
☑ Load as new layer

☐ Column(s) with unique values: geom_way   ☑ Geometry column: geom_way

Layer name (prefix): start

☐ Avoid selecting by feature id

Figure 11 : Executing query for start point



```
--find the nearest vertex to the destination longitude/latitude
destination AS (
    SELECT geom_way,topo.source --could also be topo.target
    FROM nep_2po_4pgr as topo
    ORDER BY topo.geom_way <-> ST_SetSRID(
    ST_GeomFromText('POINT (85.538562 27.619362)'),
    4326)
    LIMIT 1
```

1 rows, 0.344 seconds

	geom_way	source
1	0102000020E61...	13910

☑ Load as new layer

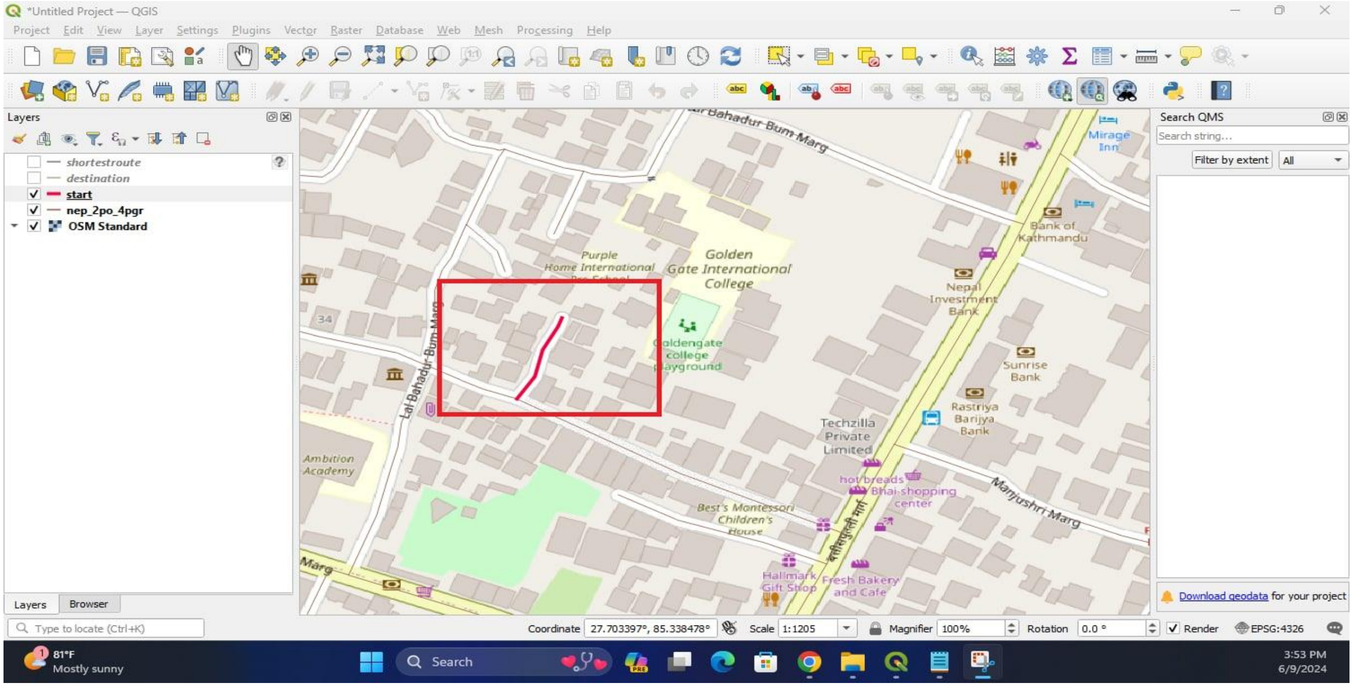
☐ Column(s) with unique values: geom_way

☑ Geometry column: geom_way

Layer name (prefix): destination

☐ Avoid selecting by feature id

Figure 12 : Executing query for destination point



☐ shortestroute

☐ destination

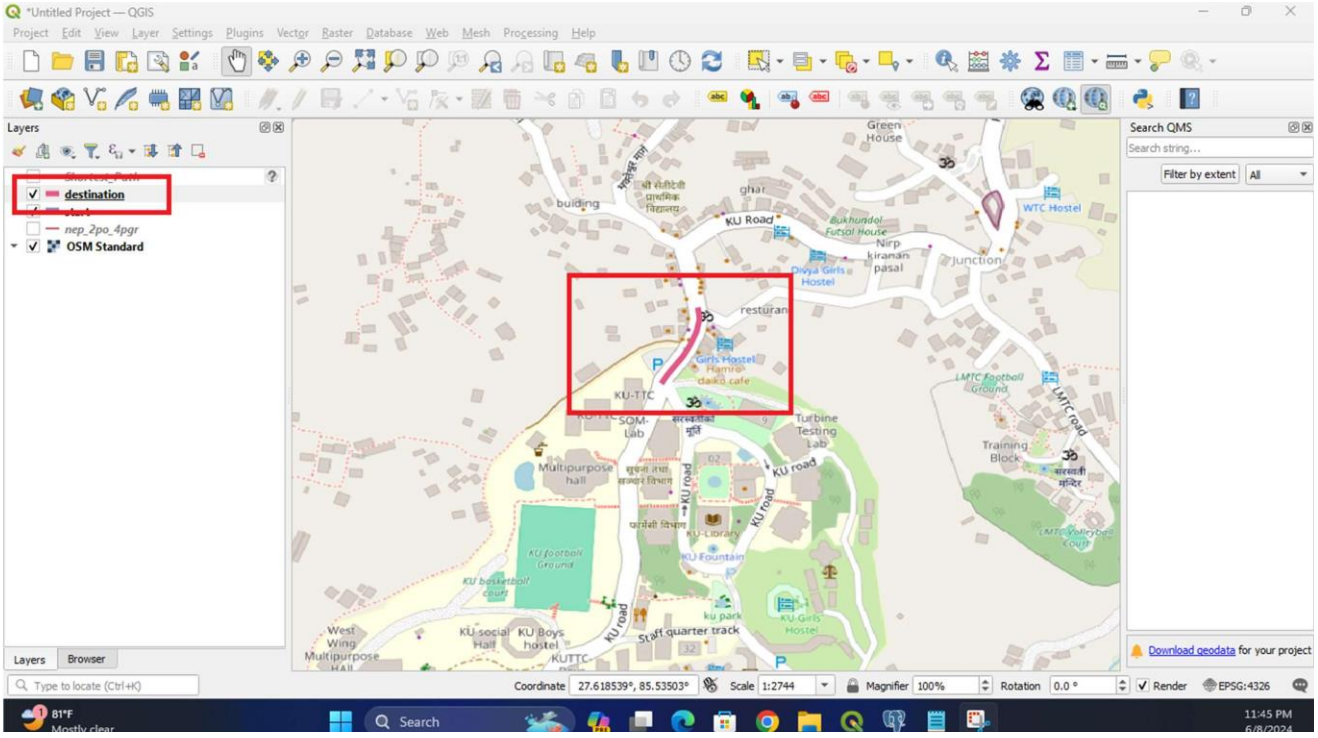
☑ start

☑ nep_2po_4pgr

☑ OSM Standard

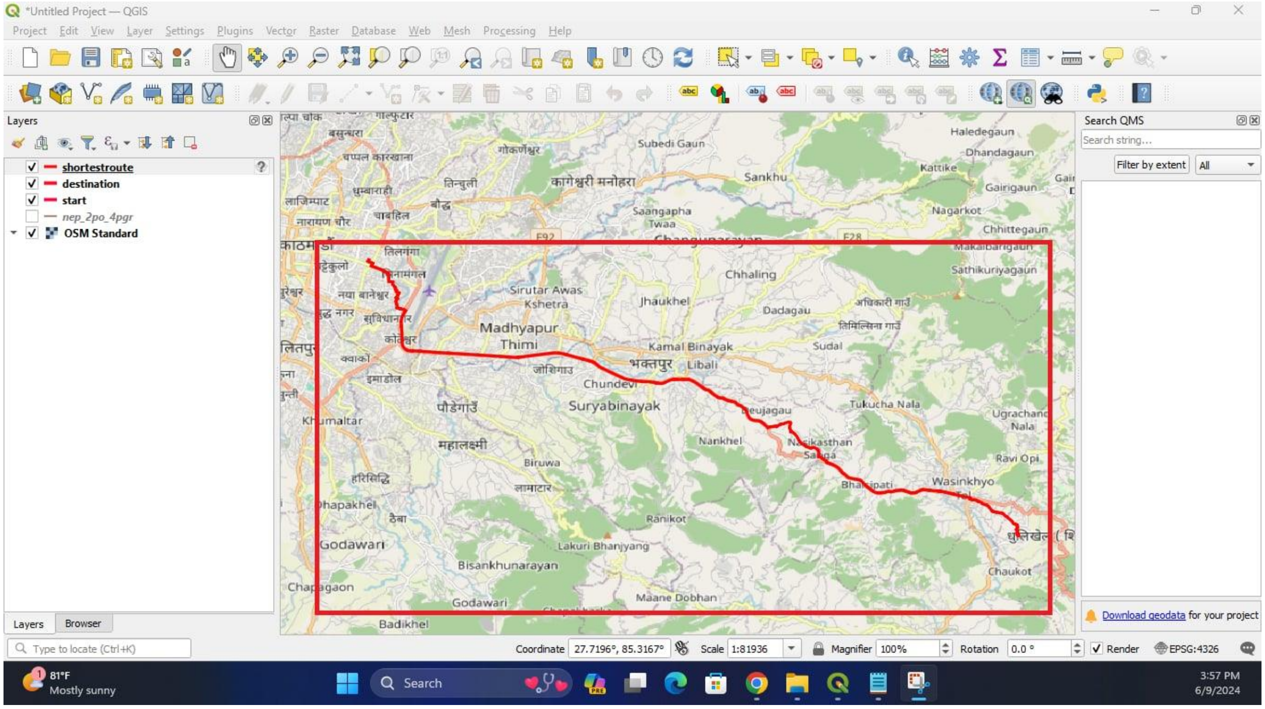
Coordinate 27.703397°, 85.338478° Scale 1:1205 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:4326

Figure 13 : Loading layer with starting point



Coordinate 27.618539°, 85.53503° Scale 1:2744 Magnifier 100% Rotation 0.0 ° Render EPSG:4326

Figure 14 : Loading layer with destination point



Coordinate 27.7196°, 85.3167° Scale 1:81936 Magnifier 100% Rotation 0.0 ° Render EPSG:4326

4. RESULTS AND DISCUSSION

The shortest route between Baneshwor and Kathmandu University was determined using pgRouting. The integration of OSM data with PostgreSQL, PostGIS, and pgRouting proved to be effective for routing purposes. The tools worked seamlessly together, enabling efficient route calculation. QGIS was used to visualize the calculated route.

5. CONCLUSION

In this project, the shortest route between Baneshwor and Kathmandu University was successfully determined using OSM data, PostgreSQL, PostGIS, and pgRouting. The process involved data preparation with osm2po, spatial analysis using pgRouting, and visualization in QGIS. The integration of these tools allowed for accurate calculation of the route and clear visualization on the map. Despite some challenges, such as data quality issues and processing time, the methodology proved effective and demonstrated