HARNESSING GIS CAPABILITY OF POSTGRESQL USING POSTGIS AND OPENSTREETMAP (OSM)

DATE : 07TH MAR 2025

TEAM: AMIT THAKUR, SARANG MAHADIK & SHRIDHAN BAKSHETTI

MICHELIN MAPPING FACTORY

OUTLINE

1. About Michelin Mapping Factory
2. Introduction to Spatial Data and PostGIS
3. OpenStreetMap (OSM) role in GIS and Tools
4. Exploring PostGIS functions
5. Interaction with GIS Data
6. Applications of GIS with PostGIS & OSM
7. Demo: OSM Insights POI Mapping
8. Demo: pgRouting shortest path finding
9. Michelin – Mapping Factory Mapmatching

MICHELIN MAPPING FACTORY

Our Mission: “Build a geospatial data platform and provide best-in-class services to support better and sustainable mobility in B2C and B2B.”

Expertise:

We provide consultancy and guidance in the geodata domain to enable and accelerate Michelin Mobility Business and beyond.

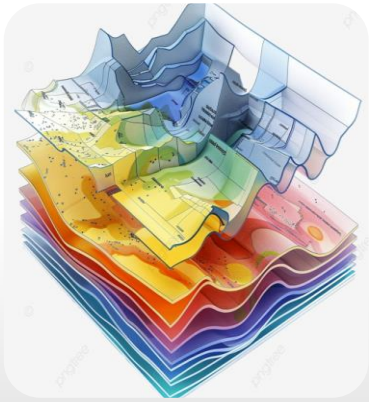
Platform:

We provide a cost-effective, sustainable, and supplier-agnostic geospatial SaaS platform to enable and accelerate Michelin Mobility Business.

Domain Expertise:

- Maps
- Routing
- Search
- Geo-contextualization

WHAT IS SPATIAL DATA



Spatial data, also known as geospatial data, refers to information that identifies the geographic location and characteristics of natural or constructed features and boundaries on the Earth.

- Points (e.g., GPS coordinates)
- Lines (e.g., roads or rivers)
- Polygons (e.g., Administrative borders – State/City, land use zones).

Used in various of application/real life use case like

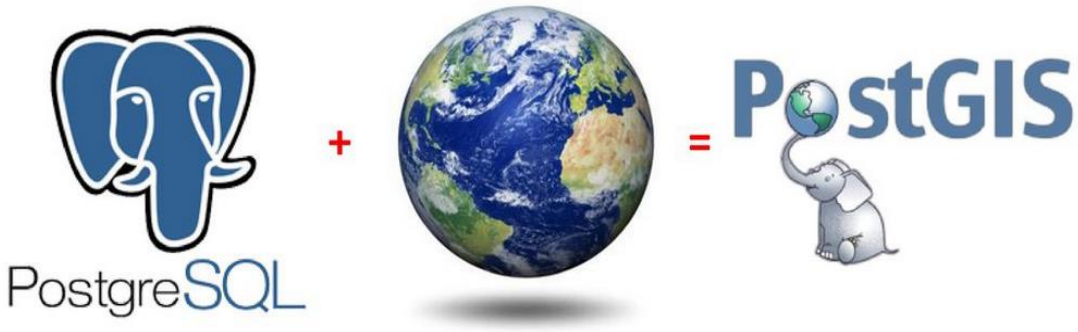
- Urban planning and infrastructure development
- Environmental management and conservation
- Disaster management and response
- Agriculture and land management
- Transportation and logistics

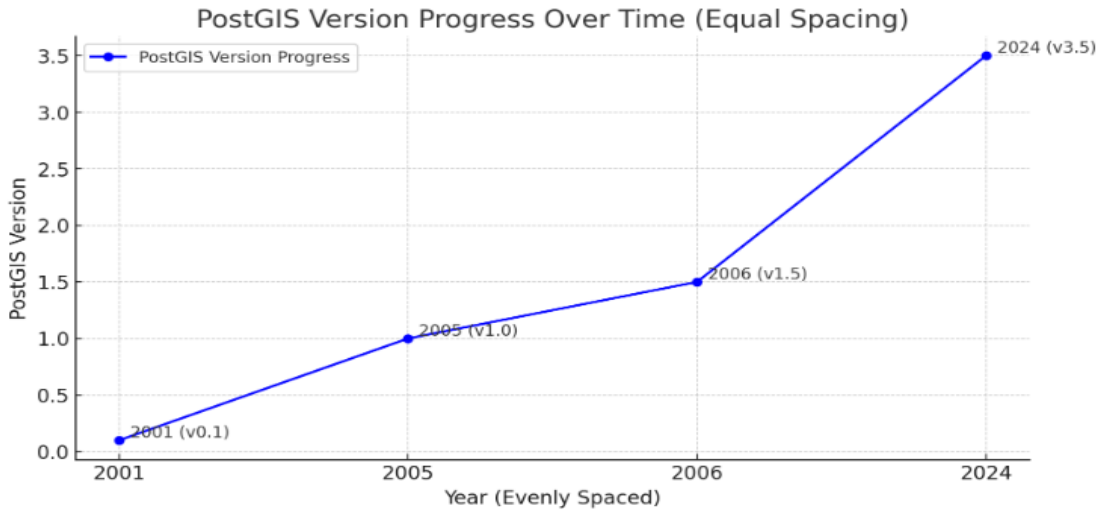
TYPE OF SPATIAL DATA

- **Vector data**
- Raster **data**
- **Attribute data**
- **Temporal data**

INTRODUCTION TO POSTGIS

- Extending PostgreSQL for spatial data support.
- Enables spatial queries and GIS functionalities.
- Complies with Open Geospatial Consortium (OGC) standards for compatibility.
- Interoperable with GIS tools like QGIS & ArcGIS
- Easy to learn for SQL users with spatial functions.
- Key formats include WKT (Well-Known Text) and GeoJSON.





PostGIS Milestone timeline

- Handles raster and vector data efficiently.
- Supports spatial joins, buffers, and distance calculation
- Provides geodetic support for working with different coordinate systems
- Used in applications like urban planning, environmental studies and navigation.

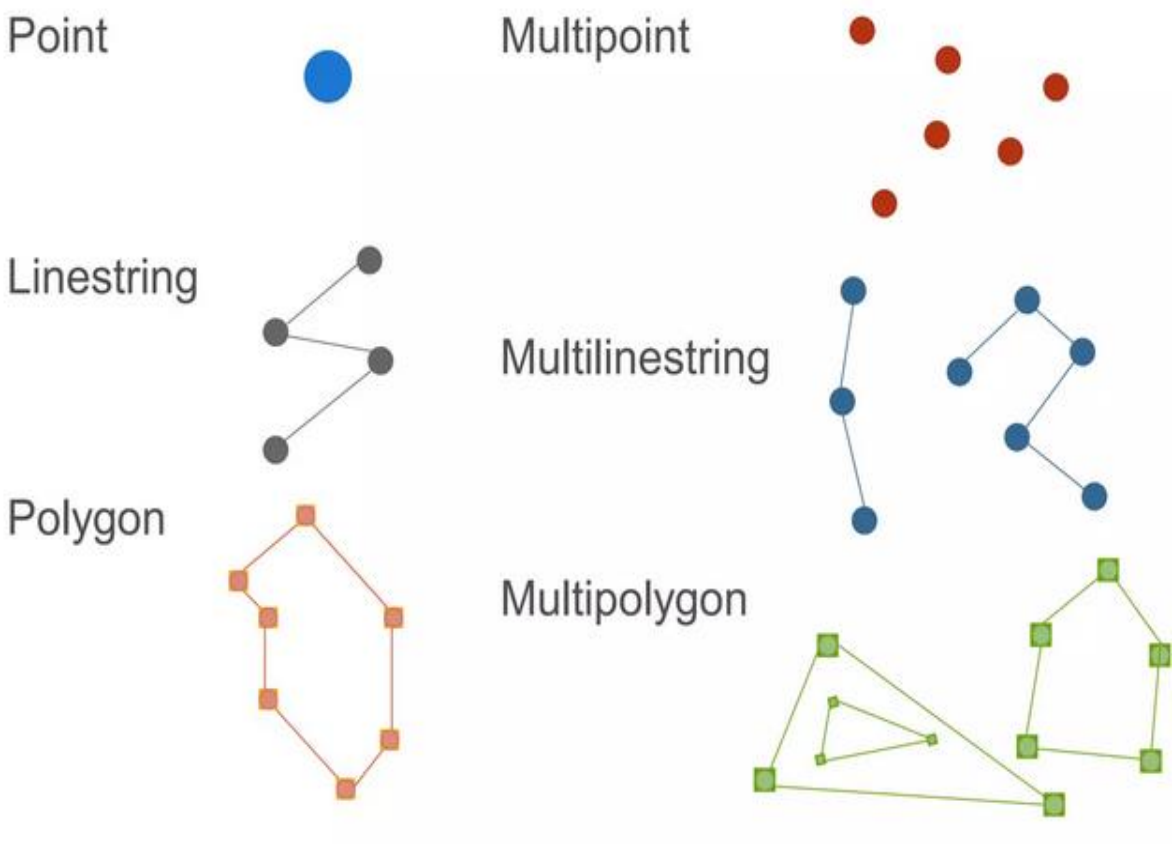
Latest Version: PostGIS 3.5

Latest Version: PostgreSQL 17.3

POSTGIS DATATYPES

Geometry column definition in a spatial database:

- `geom` → Column name
- `geometry(Type, SRID)` → Data type and spatial reference system



Examples

- geom geometry(LineString,4326)
- geom geometry(Point, 4326)
- geom geometry(Polygon, 4326)
- geom geometry(MultiPoint, 4326)
- geom geometry(MultiLineString, 4326)

OPENSTREETMAP (OSM) & GIS

- **Open-source, crowd-sourced geospatial data repository**.
  - Provides global and highly detailed map data.
  - Constantly updated and improved by users.
- **Provides roads, buildings, land use, and administrative boundaries.**
  - Data accuracy improves with contributions from the community.
  - Used in both commercial and non-commercial applications.
- **Useful for urban planning, logistics, disaster management, and navigation.**
  - Used by companies like Mapbox, Google, and government agencies.
  - Provides an alternative to proprietary mapping services.

OSM DATA AND LOAD INTO POSTGRESQL DATABASE

- **Open-source, crowd-sourced geospatial data repository.**
- **Provides roads, buildings, land use, and administrative boundaries.**
- **Useful for urban planning, logistics, disaster management, and navigation.**

LOAD DATA: install osm2pgsql osm2pgsql -d osm_db -U postgres -H localhost -P 5432 create slim cache 4000 hstore multi-geometry style yourfile.pbf

- create → Creates new tables (use append to add to existing data).
- cache 4000 → Allocates 4000MB RAM (adjust based on your system).
- hstore → Stores all OSM tags in an hstore column (useful for complex queries).

Install PostgreSQL
- Install PostGIS Extension
- Install osm2pgsql

Set Database
- CREATE DATABASE osm;
- CREATE EXTENSION postgis;

Download OSM Data

Import Data Using osm2pgsql

Generated Tables
- planet_osm_line
- planet_osm_point
- planet_osm_polygon
- planet_osm_nodes
- planet_osm_roads
- planet_osm_ways
- planet_osm_rels

OPENSTREETMAP (OSM) TOOLS

Tools for Data Processing

After downloading raw OSM data, you might need specialized tools to process or convert it.

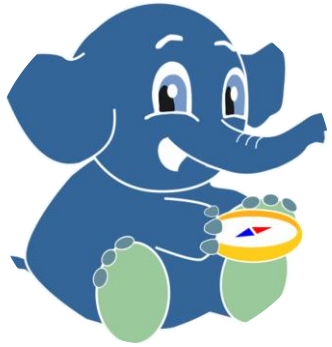
QGIS

Open-source GIS software for visualizing and analyzing OSM data. Supports plugins for importing OSM layers.

Imports OSM data into PostgreSQL/PostGIS databases for complex queries.

ogr2ogr

Converts OSM data into other geospatial formats like shapefiles or GeoJSON.



PgRouting

Advanced routing algorithms for pathfinding.

Osmosis

Command-line tool for filtering, extracting, and converting OSM data.

Useful for handling PBF or XML formats.

HOW GEOMETRY LOOK LIKE IN POSTGRESQL

select osm_id, name, way from planet_osm_linewhere osm_id = 160558221;

Well-Known Binary (WKB):

0102000020E61000001200000D3E98A636E6553408E8AA4822FF529400856D5CB6F6553406D76FFB341F529404E6CF420736553408DFB45BF6CF529408FB4
0A51746553401A51DA1B7CF52940993F4BFC76655340A31B06989EF5294026E5EE737C6553400EDC813AE5F529408F7E45C88A655340B948EBB996F629409
524743C8B655340637A67599CF6294018EC866D8B655340504BBDB89EF62940597C540D8C655340B1F84D61A5F629402FE301C08C6553409B215514AFF62

Well-Know Text (WKT) – Using ST_AsTEXT(geometry g1):

LINESTRING(77.5848626 12.9788781,77.5849485 12.9790169,77.5851519 12.9793453,77.5852244 12.9794625,77.5853873
12.9797256,77.585721 12.9802645,77.5865956 12.9816187,77.5866233 12.9816616,77.586635 12.9816797,77.5866731
12.9817305,77.5867157 12.9818045,77.5867552 12.9818726)

GeoJSON – Using ST_AsGeoJSON:

{"type":"LineString","coordinates":[[77.5848626,12.9788781],[77.5849485,12.9790169],[77.5851519,12.9793453],[77.5852244,12.9794625],[7
7.5853873,12.9797256],[77.585721,12.9802645],[77.5865956,12.9816187],[77.5866233,12.9816616],[77.586635,12.9816797],[77.5866731,12
.9817305],[77.5867157,12.9818045],[77.5867552,12.9818726],[77.5869256,12.9821676]]}

GEOMETRY INDEXES:

PostGIS provides specialized indexes to improve performance.

Syntax: *CREATE INDEX idx_way ON public.fra_diff USING GIST (**way**);*

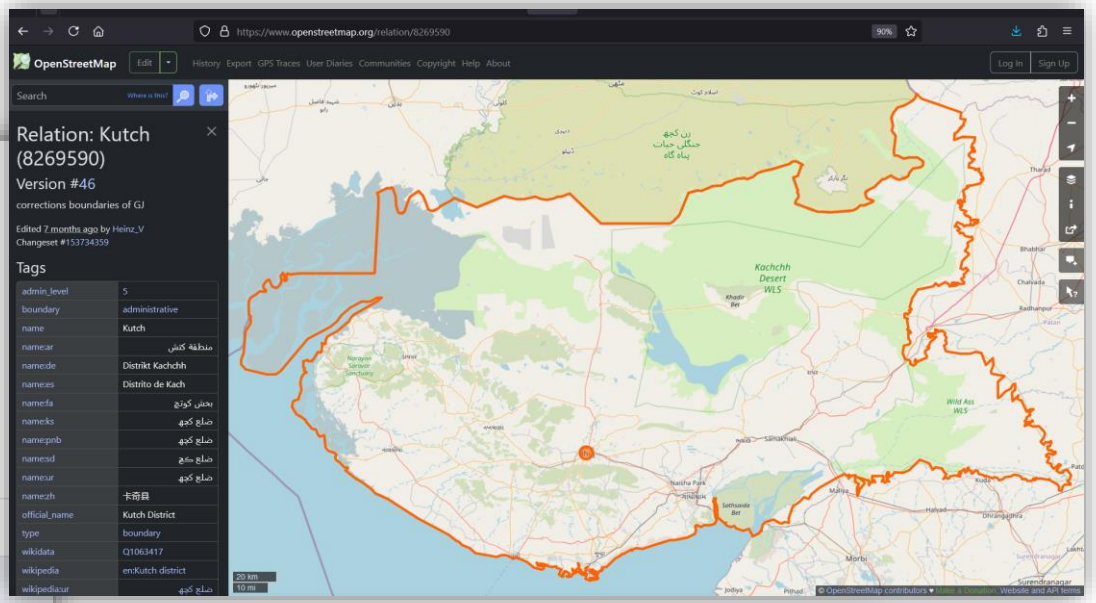
FINDING LARGEST DISTRICT – ST_AREA

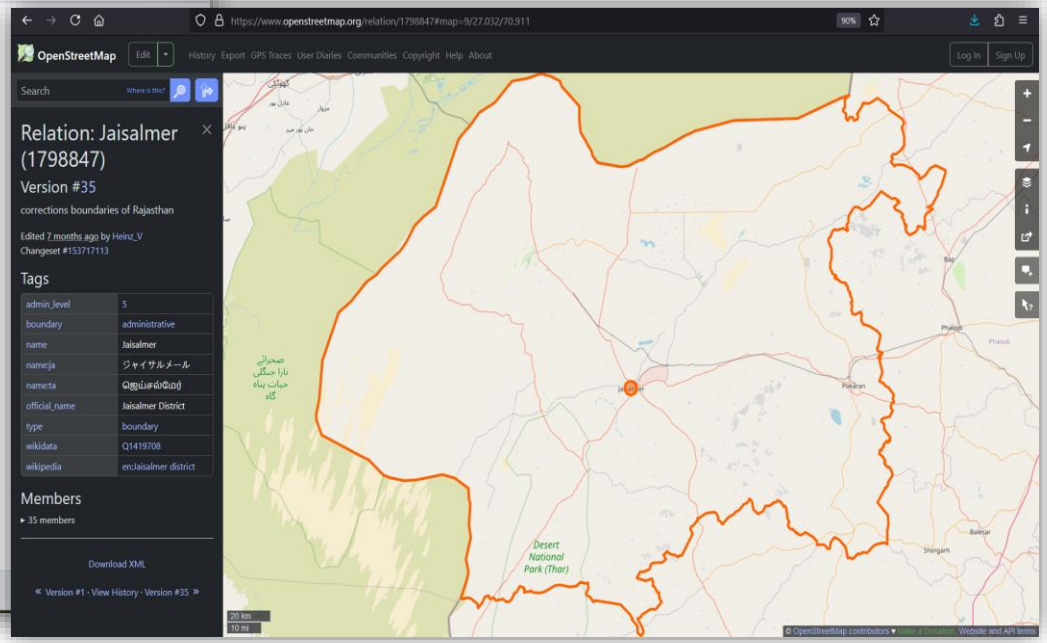
```
18
19  select osm_id, name, area_in_sq_km
20  from (select osm_id, name, st_area(way::geography)/1000000 area_in_sq_km
21          from planet_osm_polygon
22          where admin_level = '5'
23          and osm_id < 0
24      ) as z
25  order by area_in_sq_km desc
26  limit 10
27
```

Data Output | Messages | Geometry Viewer | Notifications

	osm_id bigint	name text	area_in_sq_km double precision
1	-8269590	Kutch	43076.39332639947
2	-1798847	Jaisalmer	38445.295594880175
3	-10391800	Leh district	26661.029395156904
4	-1948979	Bikaner	26109.013725298457
5	-10391802	Nubra tehsil	19947.086953780898
6	-1798824	Barmer	18213.5730122178
7	-1986123	Ahmednagar	17088.586553074005
8	-1986140	Pune	15649.756446937325
9	-1985907	Nashik	15537.765074407058
10	-1997168	Solapur	14839.04640068535

Total rows: 10 of 10 Query complete 00:00:02.101 Ln 22, Col 26





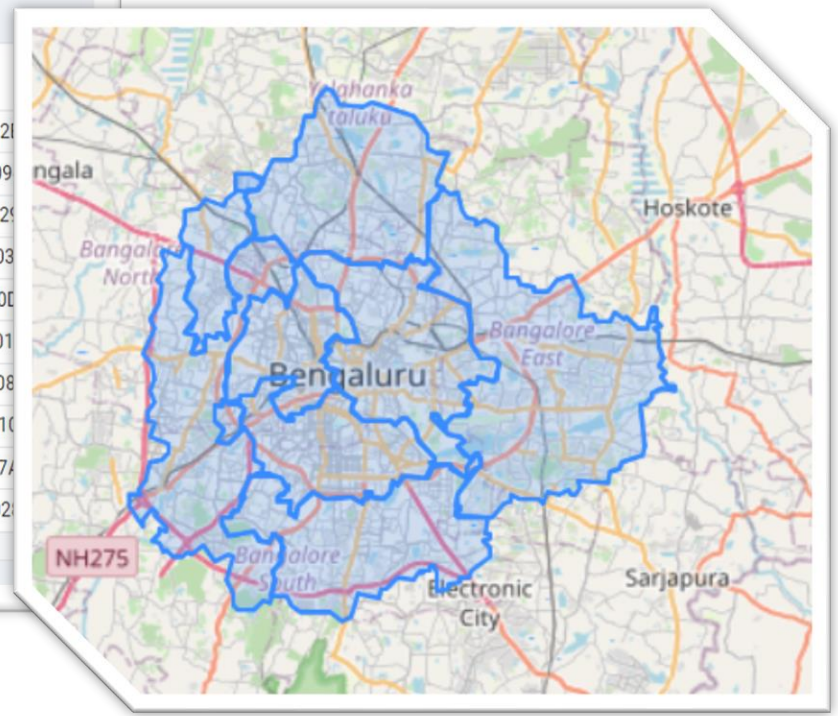
FINDING SUB-AREAS OF BENGALURU – ST_CONTAINS

```
21  select a.osm_id, a.name, b.osm_id, b.name, b.way
22  from planet_osm_polygon a
23  join planet_osm_polygon b on st_contains (a.way, b.way )
24  where a.osm_id = -7902476
25  and b.admin_level = '9'
26
27
```

Data Output | Messages | Geometry Viewer | Notifications

	osm_id bigint	name text	osm_id bigint	name text	way geometry
1	-7902476	Bengaluru	-7903994	South Zone	0103000020E6100000010000000D10500009A6615913C615340A986FD9E58EB2940F7BE02E23F6153402
2	-7902476	Bengaluru	-7903536	Bommanahalli Zone	0103000020E6100000010000000E7050000AC12769B01615340C71748ABB5C729403F366ED6056153409
3	-7902476	Bengaluru	-7903408	Mahadevapura Zone	0103000020E610000001000000088070000B309302C7F685340F0ADFEBE350E2A4098569E9B916853402
4	-7902476	Bengaluru	-7904056	East Zone	0103000020E6100000010000002606000866CC5A3F9635340C0756AD37D152A407914F8D4FB6353403
5	-7902476	Bengaluru	-7903535	Rajarajeshwari Nagar Zone	0103000020E6100000010000000ED0A00001FC07D78715D5340FC5987FE64CF2940A2F721CA725D5340
6	-7902476	Bengaluru	-7904165	West Zone	0103000020E6100000010000002206000E578AAE8B460534073BC02D193E629409DF75A2BB56053401
7	-7902476	Bengaluru	-7904166	Dasarahalli Zone	0103000020E6100000010000000A103000081DA5EC16C5F5340D8A1F54B1F1D2A4094FB7843755F53408
8	-7902476	Bengaluru	-17417779	Lakshmipura	0103000020E6100000010000004E0000002463B5F97F615340357227220D2C2A4006BD37868061534010
9	-7902476	Bengaluru	-7903409	Yelahanka Zone	0103000020E610000001000000560300005192640CE2615340F964C57075282A407A6D3656E26153407A
10	-7902476	Bengaluru	-17421305	Shigehalli	0103000020E6100000010000001C00000044E9C1934C715340933F733161032A40CD11E8024D7153402

Total rows: 10 of 10 Query complete 00:00:01.143 Ln 27, Col 1



FINDING SCHOOLS IN SUB-AREAS OF BENGALURU – ST_INTERSECTS

```
21  select a.osm_id, a.name, b.name, count(*) school_cnt
22  from planet_osm_polygon a
23  join planet_osm_polygon b on st_contains (a.way, b.way )
24  left join planet_osm_point c on ST_Intersects(b.way, c.way)
25  where a.osm_id = -7902476
26  and b.admin_level = '9'
27  and c.amenity = 'school'
28  group by a.osm_id, a.name, b.osm_id, b.name
29  ;
30
```

Data Output | Messages | Geometry Viewer | Notifications

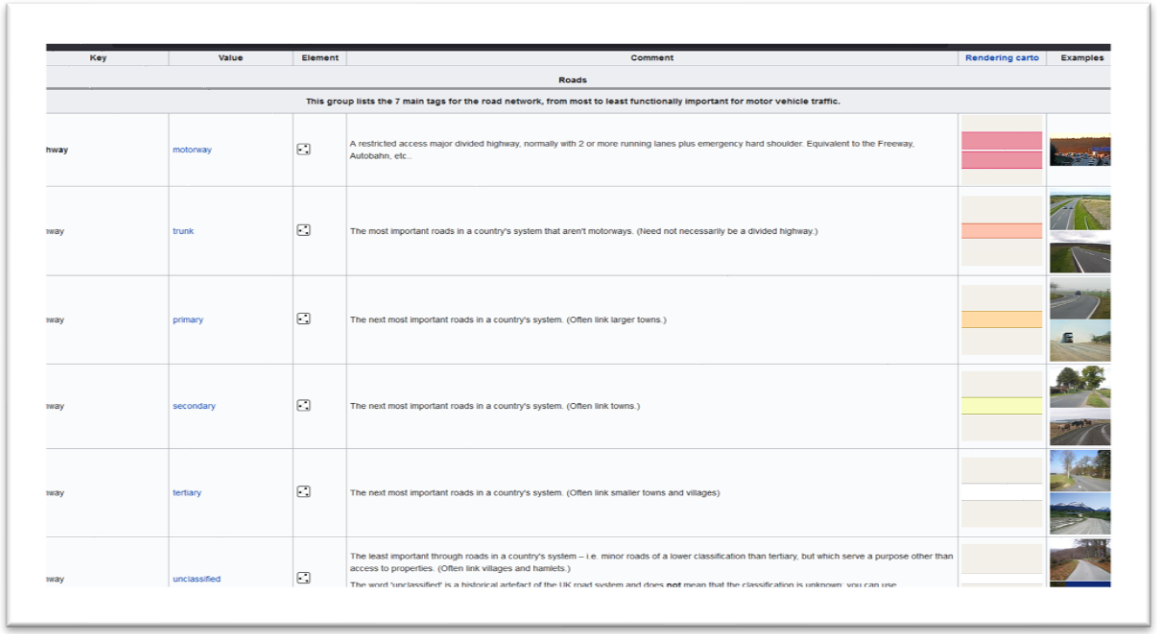
	osm_id bigint	name text	name text	school_cnt bigint
1	-7902476	Bengaluru	Dasarahalli Zone	3
2	-7902476	Bengaluru	West Zone	42
3	-7902476	Bengaluru	East Zone	122
4	-7902476	Bengaluru	South Zone	180
5	-7902476	Bengaluru	Bommanahalli Zone	172
6	-7902476	Bengaluru	Rajarajeshwari Nagar Zone	36
7	-7902476	Bengaluru	Yelahanka Zone	19
8	-7902476	Bengaluru	Mahadevapura Zone	49

TOTAL LENGTH OF ROAD BASED ON HIGHWAY TYPE – ST_LENGTH

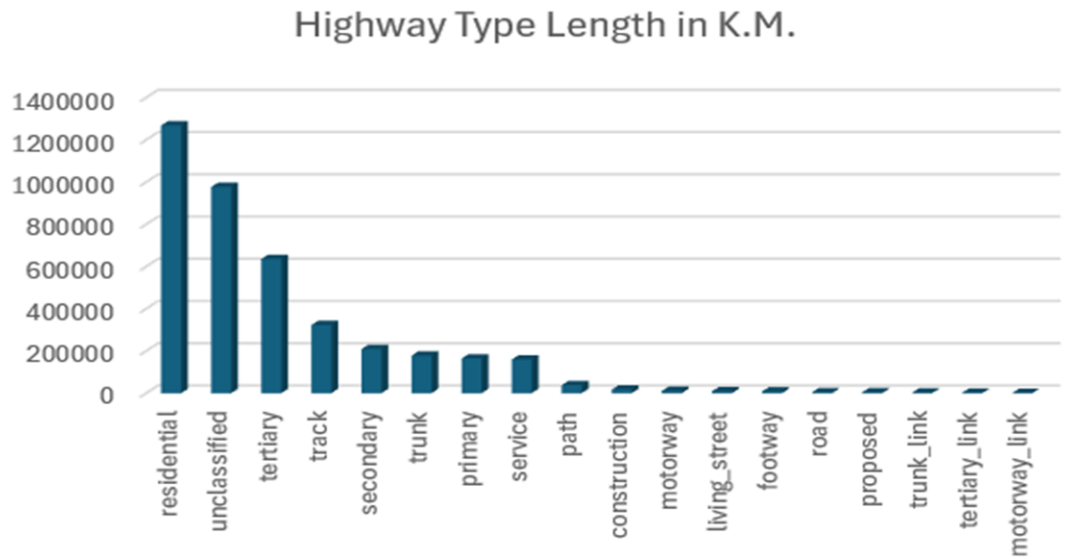
```
select  highway
        ,round(sum(st_length(way::geography)::numeric)/1000,2) len
from planet_osm_line
where highway is not null
group by 1
order by 2 desc;
```

Output | Messages | Geometry Viewer | Notifications

highway text	len numeric
residential	1268286.91
unclassified	976709.60
tertiary	635687.35
track	322606.74
secondary	208678.30
trunk	177615.36
primary	164309.71
service	159142.56
path	37901.61
construction	17017.47
motorway	10844.53
living_street	8779.46
footway	8108.71
road	4401.69
proposed	4371.24
trunk_link	2806.08
tertiary_link	2347.96



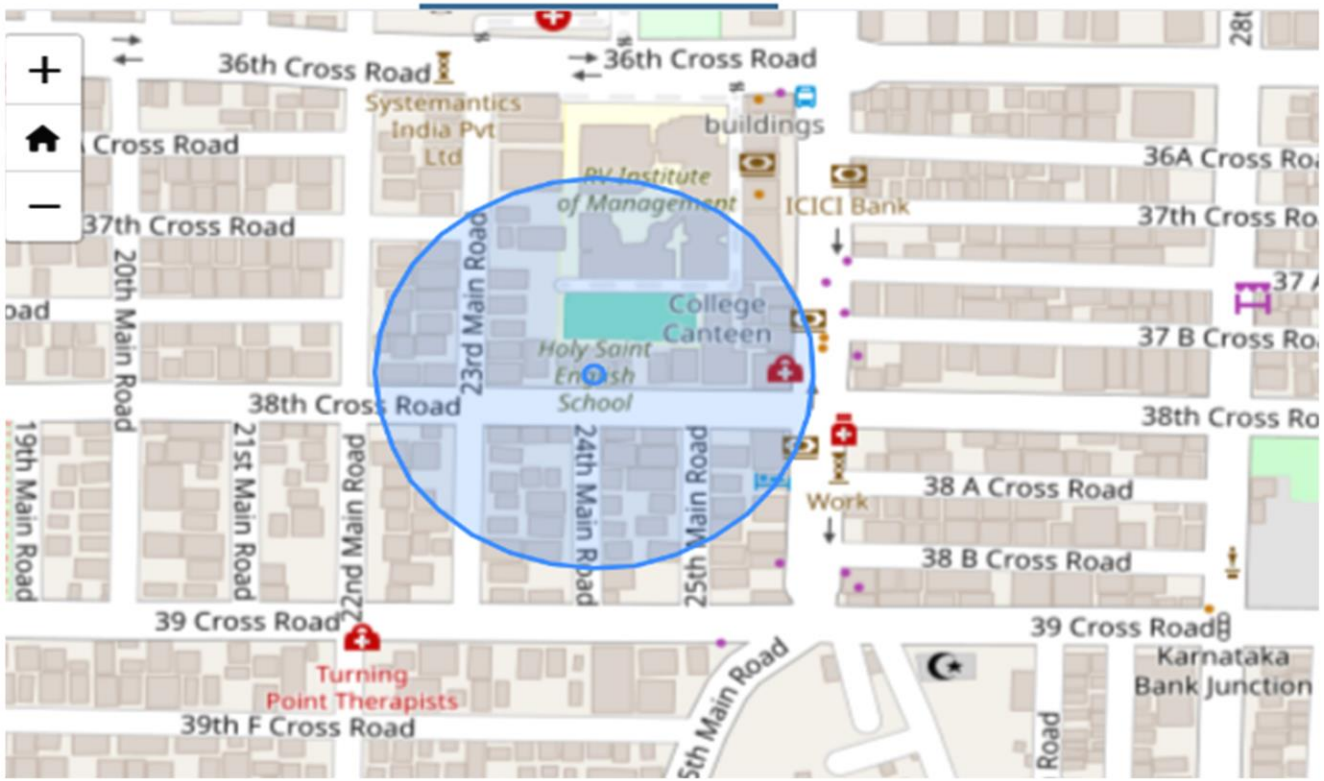
Key	Value	Element	Comment	Rendering carto	Examples
Roads					
This group lists the 7 main tags for the road network, from most to least functionally important for motor vehicle traffic.					
hway	motorway		A restricted access major divided highway, normally with 2 or more running lanes plus emergency hard shoulder. Equivalent to the Freeway, Autobahn, etc..		
way	trunk		The most important roads in a country's system that aren't motorways. (Need not necessarily be a divided highway.)		
way	primary		The next most important roads in a country's system. (Often link larger towns.)		
way	secondary		The next most important roads in a country's system. (Often link towns.)		
way	tertiary		The next most important roads in a country's system. (Often link smaller towns and villages)		
way	unclassified		The least important through roads in a country's system – i.e. minor roads of a lower classification than tertiary, but which serve a purpose other than access to properties. (Often link villages and hamlets.) The word 'unclassified' is a historical artefact of the UK road system and does not mean that the classification is unknown; you can use		



ADDING NO HORN ZONE AROUND SCHOOL - ST_BUFFER

```
10
11  select name, st_buffer(way::geography, 100)
12  from planet_osm_point
13  where osm_id = 1015924248
```

Data Output | Messages | Geometry Viewer | Notifications



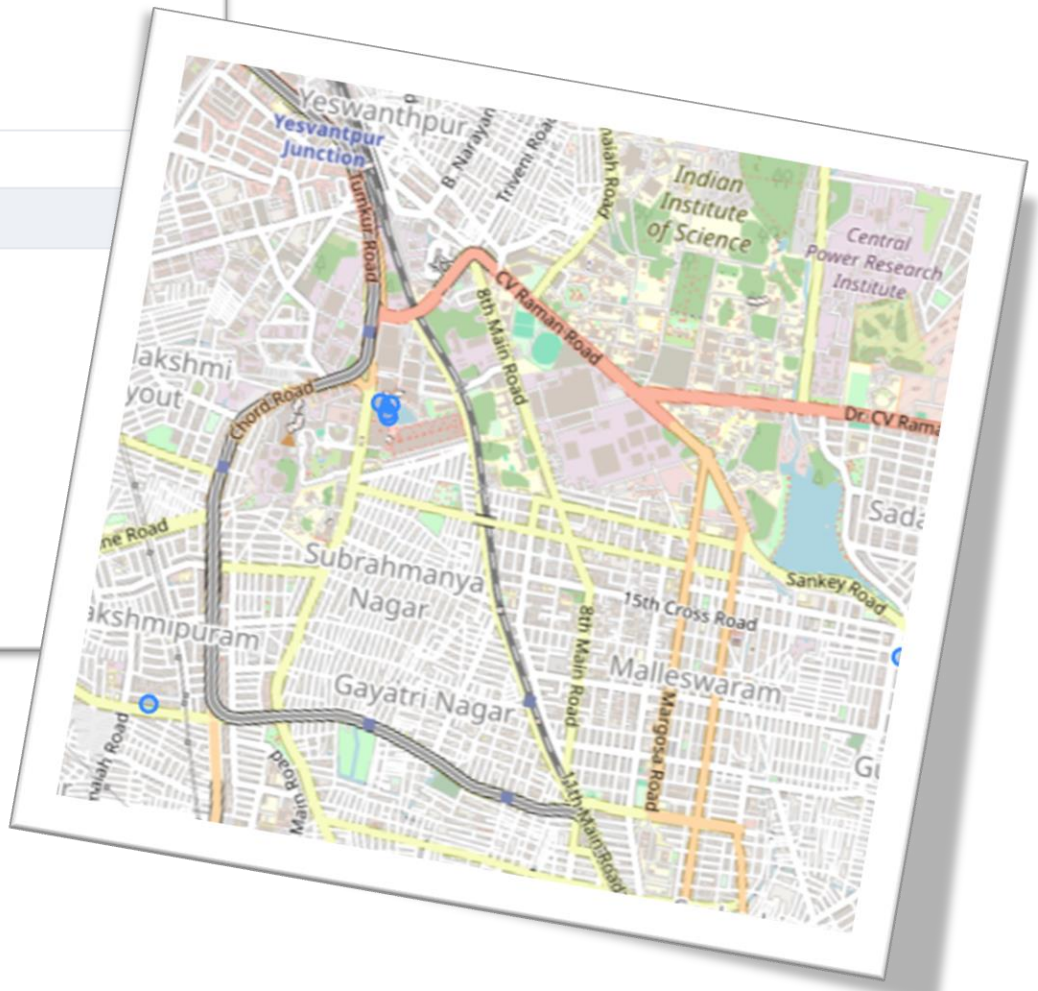
FINDING RESTAURANTS AROUND PGCONF 2025 LOCATION – ST_DWITHIN

```
125
126  select b.name,  tags -> 'cuisine', b.way
127  from (select st_setsrid(st_makepoint(77.554985,13.012627), 4326)::geography pg_conf_25_loc) a
128  join planet_osm_point b on st_dwithin(a.pg_conf_25_loc , b.way::geography, 5000)
129  where b.amenity = 'restaurant'
130  and tags -> 'cuisine' in ( 'japanese', 'italian', 'french', 'mexican');
131
```

Data Output | Messages | Geometry Viewer | Notifications

SQL

	name text	?column? text	way geometry
1	California Burrito	mexican	0101000020E6100000E8363C180C6553407BF3C06CA7022A40
2	Onesta	italian	0101000020E610000019517F62096353400B1D8535DFFE2940
3	California Burrito	mexican	0101000020E61000003F50132285635340AA3AF59210062A40
4	You Mee	japanese	0101000020E6100000835AC2908B63534028EC472F0F062A40
5	Toscano	italian	0101000020E61000008976BA988B635340B25EB12BD2052A40
6	Cafe Noir	french	0101000020E6100000C5ABAC6D8A6353408C65FA25E2052A40



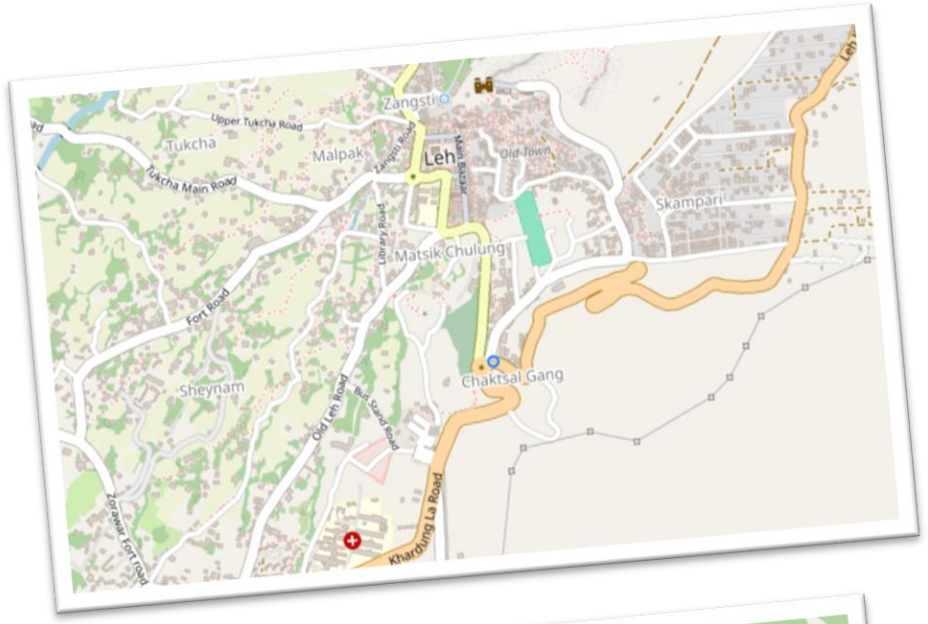
FINDING ELEVATION FOR LAT/LON USING RASTER DATA AND ST_VALUE

Elevation for given point in LEH

```
select x_ref, y_ref, ST_Value(rast, 1, ST_SetSRID(ST_Point( 77.5860235, 34.1584637 ),4326)) alt
from    mnt.MF_GeoTiff_tiles_90 r
where   x_ref = '77'
and     y_ref = '34';
```

Data Output | Messages | Geometry Viewer | Notifications

	x_ref smallint	y_ref smallint	alt double precision	st_setsrid geometry
1	77	34	3492	0101000020E6100000FFCBB56881655340348DDC8948144140

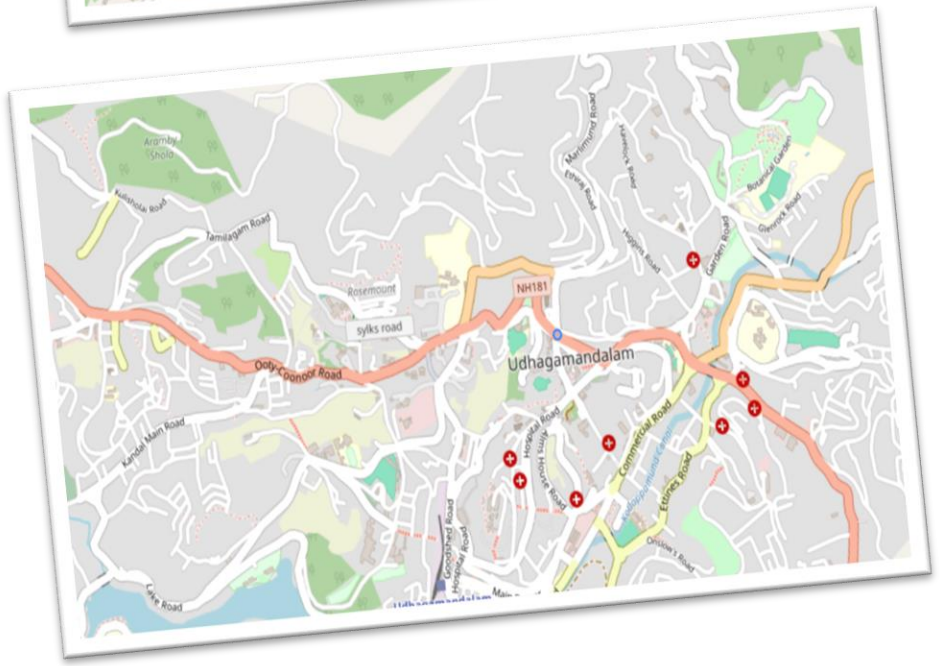


Elevation for given point on OOTY-Coonoor Road

```
select  x_ref, y_ref, ST_Value(rast, 1, ST_SetSRID(ST_Point( 76.7025312, 11.4135158 ),4326)) alt
from    mnt.MF_GeoTiff_tiles_90 r
where   x_ref = '76'
and     y_ref = '11';
```

Data Output | Messages | Geometry Viewer | Notifications

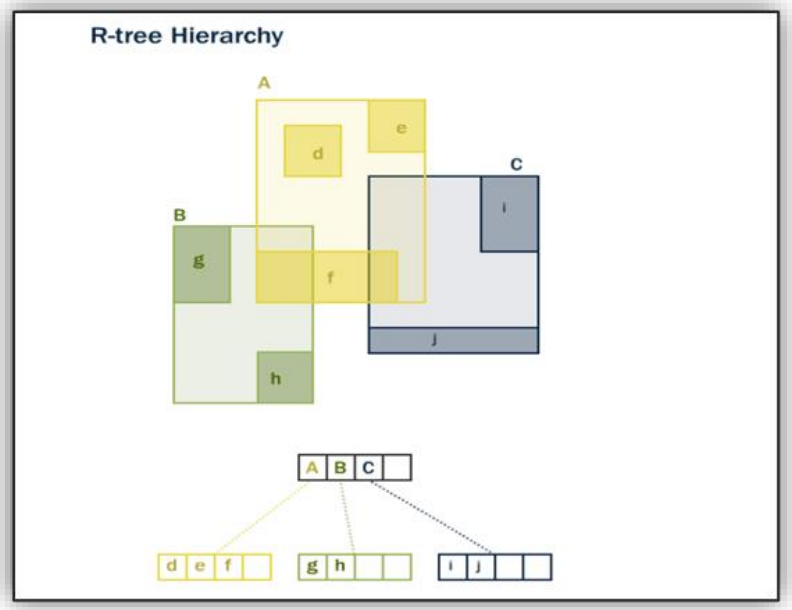
	x_ref smallint	y_ref smallint	alt double precision	pt geometry
1	76	11	2262	0101000020E6100000DB1A6C45F62C534031C2CA57B8D32640

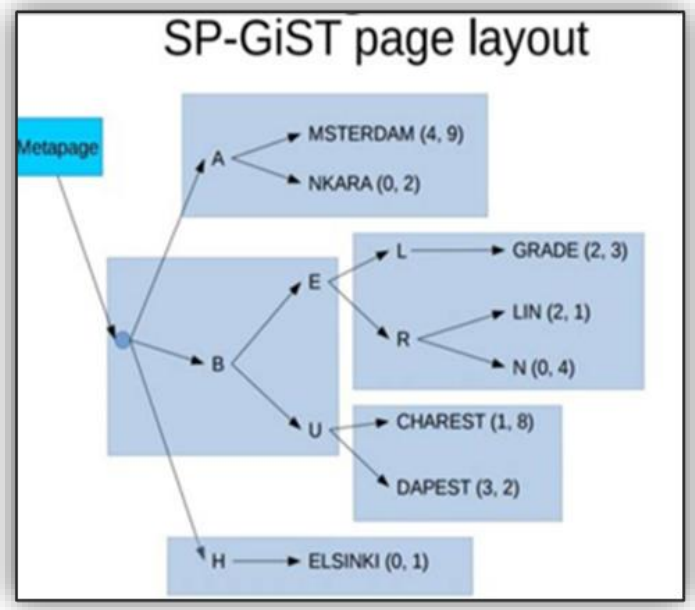


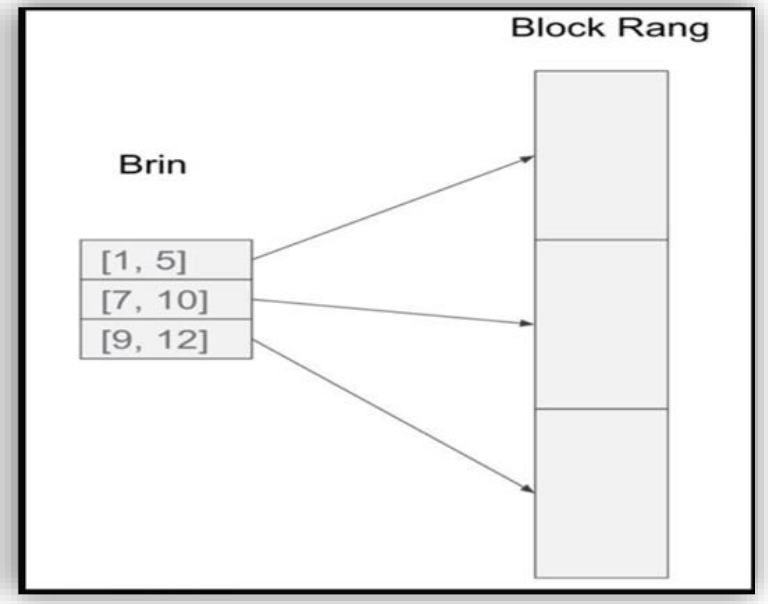
OPTIMIZING SPATIAL QUERIES WITH POSTGIS INDEXES

- Spatial queries on large datasets can be slow without indexing.
- PostGIS provides specialized indexes to improve performance.
- Choosing the right index depends on query type & dataset size.

Query Type	Best Index
Find all parks in a city	GIST
Find the closest hospital	SP-GiST
Analyze large raster datasets	BRIN
Filter by city names	B-Tree



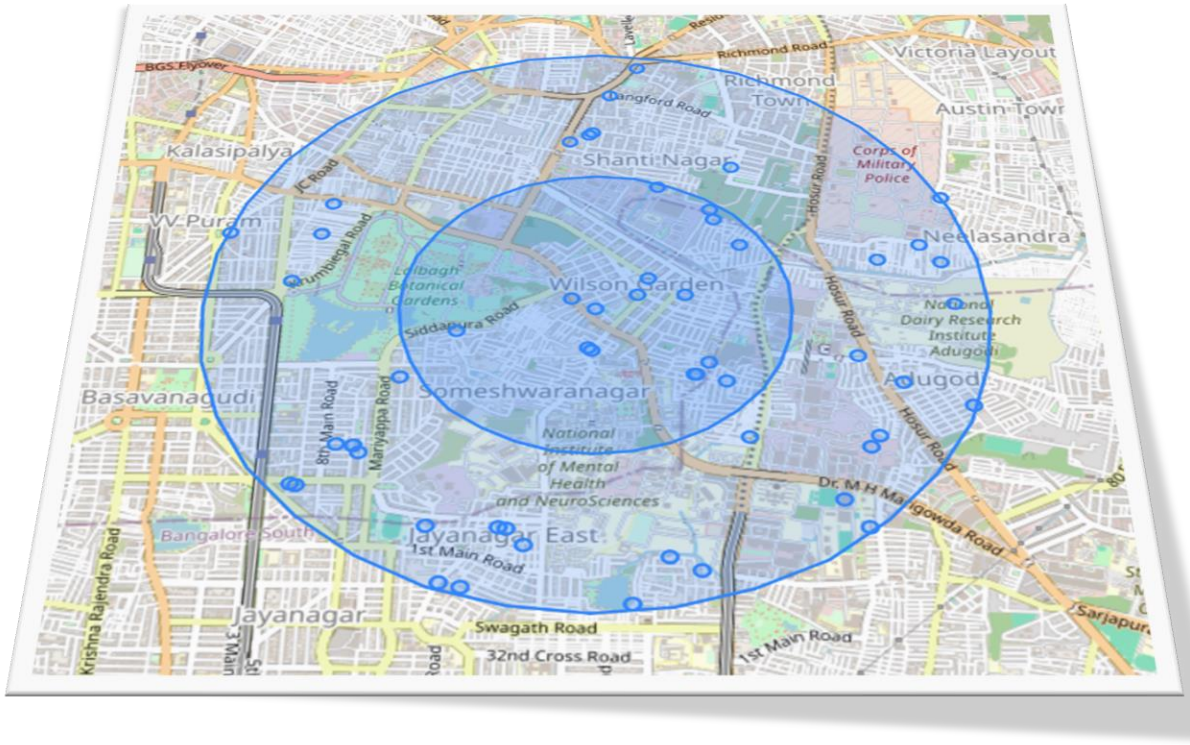




OSM INSIGHTS POI MAPPING: SCHOOLS

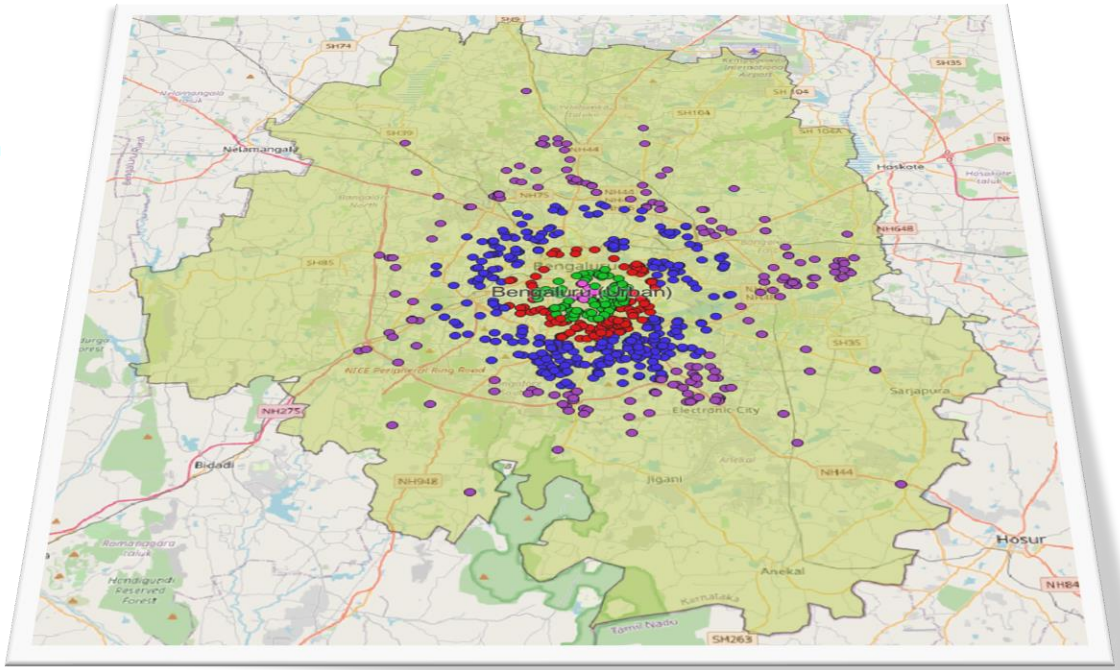
- GeoFabrik: Download PBF *(https://download.geofabrik.de/)*
- Osmium: Clip file and extract contains : *osmium extract -b <min_lon>,<min_lat>,<max_lon>,<max_lat> input.pbf -o output.pbf*
- PostgreSQL/PostGIS: Extract POIs and update based on distance
- QGIS: Display data for visualization and data analysis

School Near Bangalore City Center



10km
1km
3km
5km
Beyond 10km

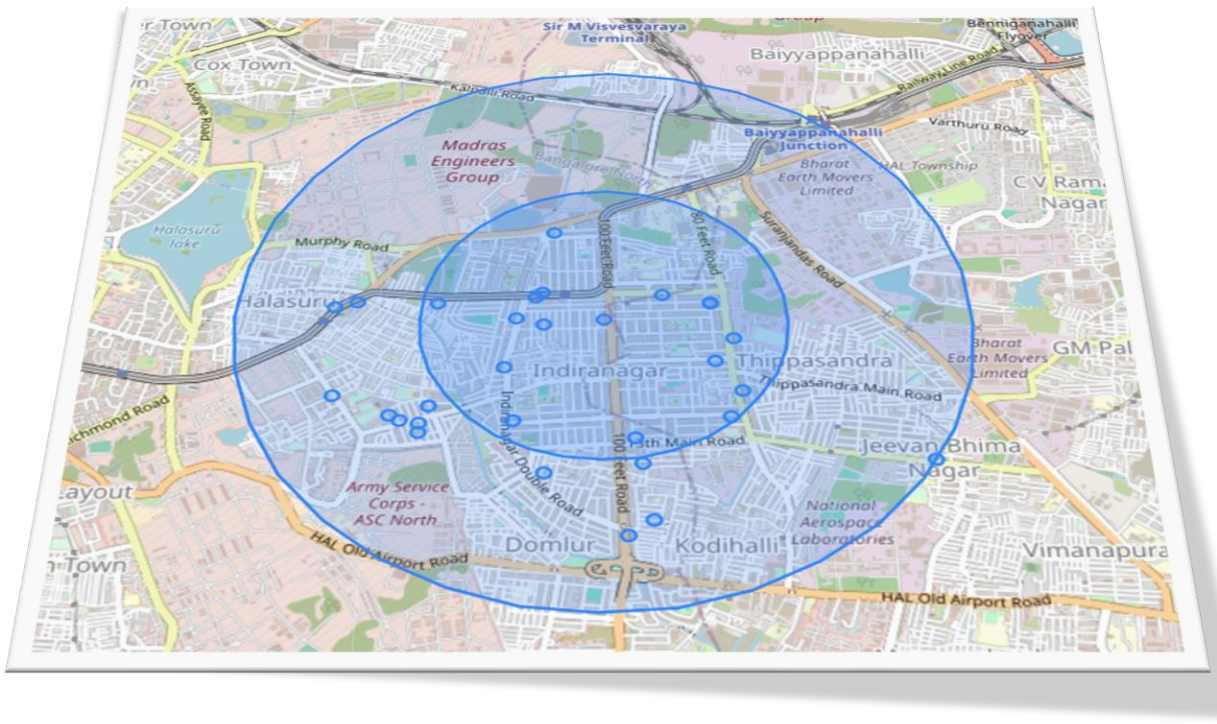
Schools Near Bangalore City Center

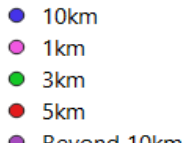


OSM INSIGHTS POI MAPPING: HOSPITALS

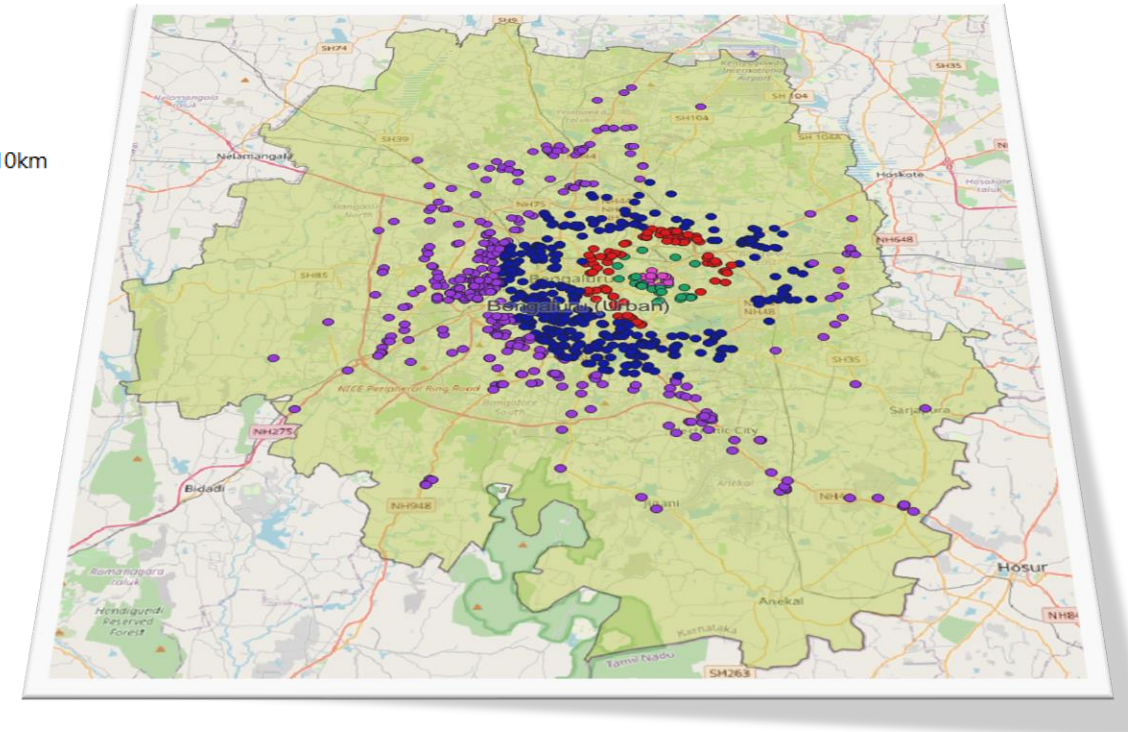
- GeoFabrik: Download PBF
- Osmium: Clip file and extract contains
- PostgreSQL/PostGIS: Extract POIs and update based on distance
- QGIS: Display data for visualization and data analysis

Hospitals Near Indiranagar, Bangalore





Hospitals Near Indiranagar, Bangalore



PGROUTING : SHORTEST PATH DEMO

PgRouting:

- Extends PostGIS for network routing
- Enables shortest path and route optimization

Dijkstra's algorithm using **PgRouting** to find the shortest path between two nodes in the road network

`pgr_dijkstra` → This function implements **Dijkstra's algorithm**, a shortest path algorithm that finds the least-cost route between two nodes.

SQL query runs Dijkstra's algorithm:

- **Input**: Find the shortest path from **node 1 to node 10**
- **Output**: A step-by-step route showing which edges (road segments) to take and the total cost (e.g., distance)

*SELECT * FROM pgr_dijkstra(*
'SELECT id, source, target, cost FROM lux_lines', 1, 10, false);

seq integer	path_seq integer	node bigint	edge bigint	cost double precision	agg_cost double precision
1	1	1	1	160.851442591025	0
2	2	2	1406	78.19927926498181	160.851442591025
3	3	3	2	197.02384106629282	239.05072185600682
4	4	4	3029	354.0082873656164	436.0745629222996
5	5	10	-1	0	790.082850287916

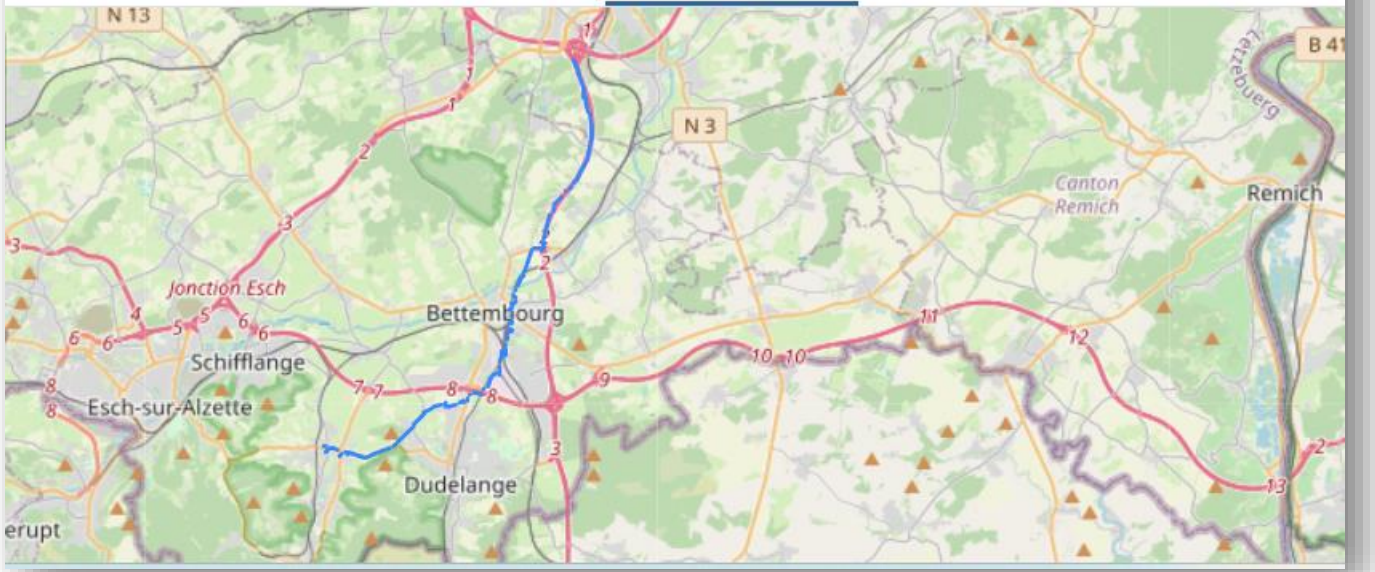
Shortest path details

PGROUTING : SHORTEST PATH DEMO VISUALIZATION

The shortest path between **node 10 and node 800** using **PgRouting's Dijkstra algorithm**.

```
SELECT geom
FROM lux_lines
WHERE id IN (
    SELECT edge FROM pgr_dijkstra(
        'SELECT id, source, target, cost FROM lux_lines',10, 800, false));
```

Output Explain Messages Notifications Geometry Viewer



Itinerary: Rue Denis Netgen, Kayl --> A3, Croix de Gasperich, Luxembourg

PgRouting provides several **graph algorithms** for **shortest path, routing, and network analysis**.

- **pgr_bdDijkstra** – Bidirectional Dijkstra, faster than pgr_dijkstra
- **pgr_contractionHierarchy** – Uses preprocessed data to speed up shortest path queries in large road networks
- **pgr_astar** – A heuristic-based shortest path algorithm, faster than Dijkstra for large road networks.
- **pgr_floydWarshall** – Computes shortest paths between all node pairs, suitable for small networks.
- **pgr_turnRestrictedPath** – Computes shortest paths while considering turn restrictions in road networks.

APPLICATIONS OF GIS WITH POSTGIS & OSM

- **Understanding driving patterns and mobility trends.**
  - Analyzing traffic congestion and optimizing routes.
  - Enhancing smart city initiatives.

- **Intelligence gathering about locations and infrastructure.**
  - Identifying key landmarks and geographic insights.
  - Assisting in security and emergency response planning.

- **Administrative boundary generation and geospatial hierarchy creation.**
  - Helps define city and country boundaries.
  - Supports electoral mapping and zoning regulations.

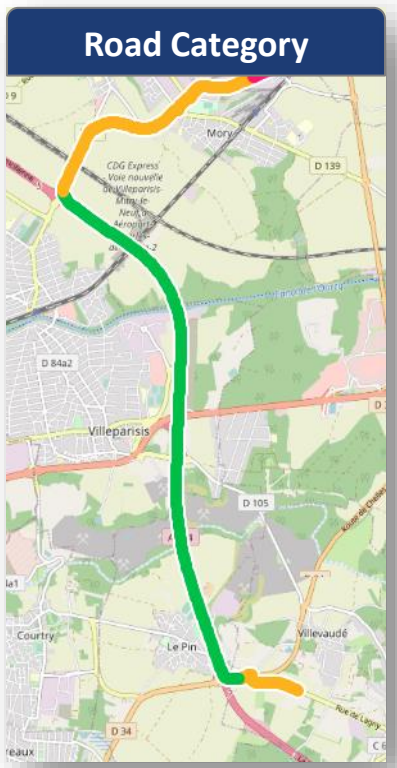
- **Integration with web maps and GIS applications.**
  - Used in real-time mapping applications and dashboards.
  - Enables interactive GIS web applications.

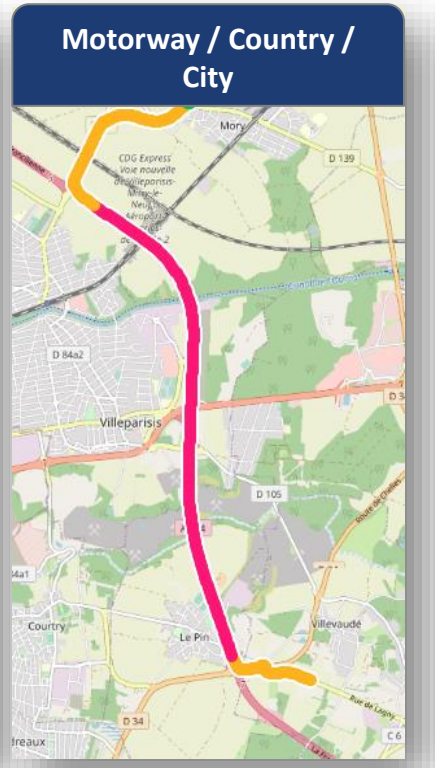
MICHELIN - MAPPING FACTORY MAPMATCHING

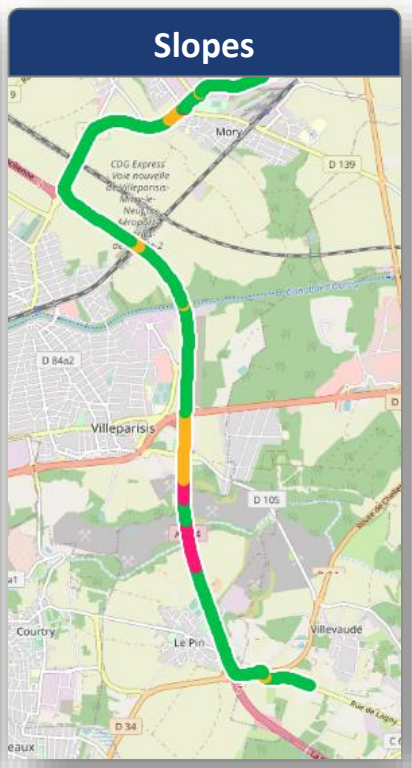
"An API to add context information to GPS tracks to extract more value from them. It is a powerful tool for data scientists. It simplifies geographical data analysis and scales from the POC stage up to a Big Data production service."

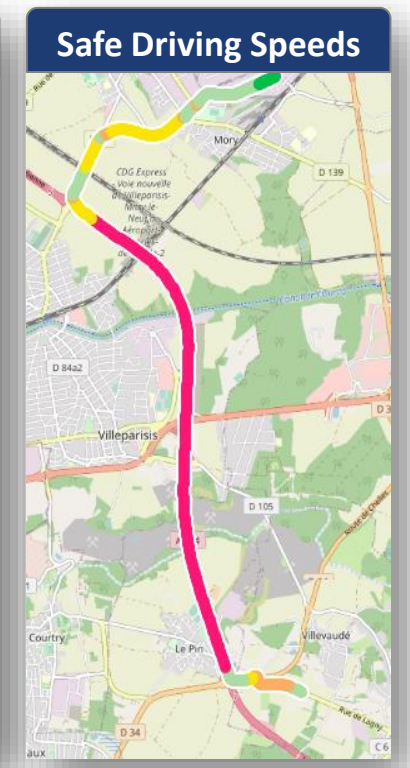
MAP-MATCHING

ROAD CONTEXTUALIZATION

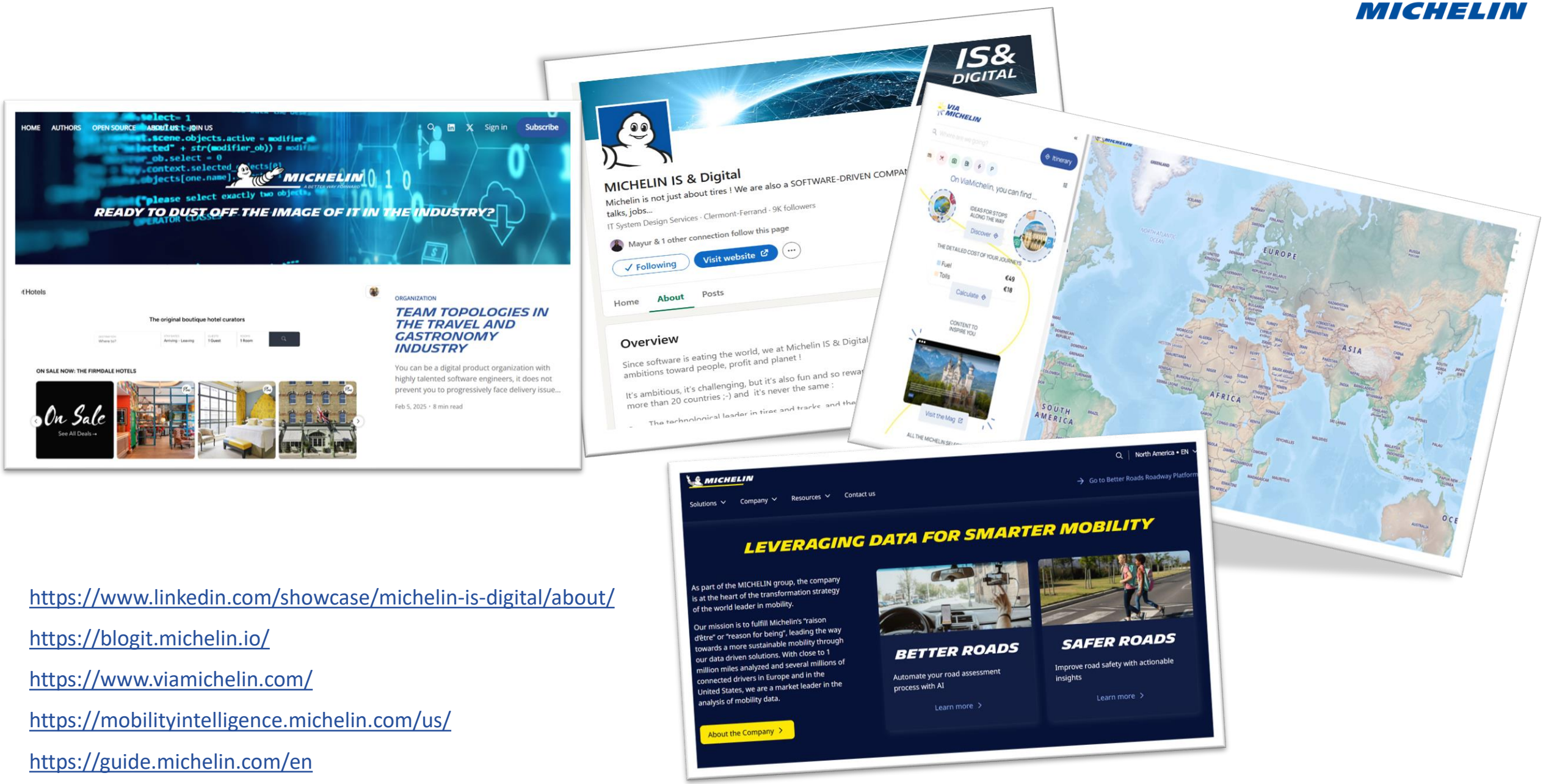








Michelin is not just about tires ! We are also a SOFTWARE-DRIVEN COMPANY



https://www.linkedin.com/showcase/michelin-is-digital/about/

https://blogit.michelin.io/

https://www.viamichelin.com/

https://mobilityintelligence.michelin.com/us/

https://guide.michelin.com/en

THANK YOU !

FROM

MICHELIN MAPPING FACTORY TEAM