Package: roadnetdb (via r-universe)

October 31, 2024

Title Package for road network operation with pgRouting

Version 0.0.4

Description This package provides functions for road network operation with pgRouting.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports cli, DBI, dplyr, glue, osmdata, purrr, rlang, RPostgres, sf, tidyr

Depends R (>= 4.1.0)

LazyData true

Repository https://nononoexe.r-universe.dev

RemoteUrl https://github.com/NONONOexe/roadnetdb

RemoteRef HEAD

RemoteSha cbe8178c9c2b2739f6a078819c443578a5c11220

Contents

`connect_database` *Connect to the database*

Description

Connect to the database. To disconnect, call `disconnect_database()`.

Usage

```
connect_database(
  user = "admin",
  password = "p@ssw0rd",
  host = "localhost",
  port = 5432,
  dbname = "roadnetdb"
)
```

Arguments

`user`	The user name (default: "admin").
`password`	The password (default: "admin").
`host`	The host name (default: "localhost").
`port`	The port number (default: 5432).
`dbname`	The database name (default: "roadnetdb").

Value

A connection to the database.

Examples

```
## Not run:
connect_database()

## End(Not run)
```

`create_table_osm_roads` *Create a table of OpenStreetMap road*

Description

Create a table named `osm_roads` in the database. The `osm_roads` manages OpenStreetMap road data and has the following columns:

- `osm_id`: The ID of the road in OpenStreetMap
- `road_type`: The type of the road
- `road_name`: The name of the road
- `layer`: The layer of the road. The layer represents the height of the road.
- `oneway`: Whether the road is one-way or not
- `geom`: The geometry of the road

Usage

```
create_table_osm_roads(conn)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.

Value

The function invisibly returns `TRUE` if the table was successfully created, `FALSE` otherwise.

Examples

```
## Not run:
create_table_osm_roads(con)

## End(Not run)
```

`disconnect_database` *Disconnect from the database*

Description

Disconnect from the database.

Usage

```
disconnect_database(conn)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.

Value

The function returns `TRUE` if this is connected and disconnecting was done as expected or `FALSE` if this is not connected to the database.

Examples

```
## Not run:
disconnect_database(con)

## End(Not run)
```

`download_osm_roads` *Download road data from OpenStreetMap*

Description

The function download road data from OpenStreetMap using overpass query. Roads are records have the following values for the key, 'highway' in OpenStreetMap data.

- motorway
- trunk
- primary
- secondary
- tertiary
- unclassified
- residential
- motorway_link
- trunk_link

- primary_link
- secondary_link
- tertiary_link

For one-way roads, the column oneway is `TRUE`.

Usage

```
download_osm_roads(area)
```

Arguments

`area`	A `sf`, area to be downloaded

Value

OpenStreetMap road data

Examples

```
## Not run:
download_osm_roads(toyota)

## End(Not run)
```

`enable_extension` *Enable database extensions.*

Description

Enable database extensions.

Usage

```
enable_postgis(conn)

enable_pgrouting(conn)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.

Value

The function returns `FALSE` if enabling fails or `TRUE` if enabling succeeds.

Examples

```
## Not run:
enable_postgis(con)

enable_pgrouting(con)

## End(Not run)
```

`fetch_osm_roads` *Fetch OpenStreetMap road data from the database*

Description

The function fetch road data intersecting with the specified area from `osm_roads` table in the database.

Usage

```
fetch_osm_roads(conn, area)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.
`area`	A `sf`, area to be fetched.

Format

A sf with 6 variables:

osm_id The ID of the road in OpenStreetMap

road_type The type of the road

road_name The name of the road

layer The layer of the road. The layer represents the height of the road.

oneway Whether the road is one-way or not

geom The geometry of the road

Value

OpenStreetMap road data.

Examples

```
## Not run:
fetch_osm_roads(con, toyota)

## End(Not run)
```

`generate_road_network` *Generate road network from road data*

Description

The function generates a road network in the specified area. A road network has intersection as nodes and road segments as edges. The function creates two tables representing edges and nodes of the network.

The function generates a road network by the following steps:

1. Generate road network edges from road data
2. Generate road network nodes from road network edges
3. Store the road network edges and nodes into the database

The road network edge table has the following columns:

- `id`: The ID of the edge
- `source`: The ID of the source node
- `target`: The ID of the target node
- `oneway`: Whether the edge is one-way or not
- `length`: The length of the edge in meters
- `parent`: The ID of the original road in OpenStreetMap
- `geom`: The geometry of the edge

The road network nodes table has the following columns:

- `id`: The ID of the node
- `geom`: The geometry of the node

The nodes are computed by `pgr_createTopology` function in pgRouting. If the distance between two nodes is less than 1 meter in Web Mercator (EPSG 3857), the two nodes are merged into one node.

Usage

```
generate_road_network(
  conn,
  area,
  node_table = "road_nodes",
  edge_table = "road_edges"
)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.
`area`	A `sf`, area to be computed.
`node_table`	A `character`, the name of the road network node table (default: `road_nodes`).
`edge_table`	A `character`, the name of the road network edge table (default: `road_edges`).

Value

Number of rows of data updated.

Examples

```
## Not run:
generate_road_network(con, toyota)

## End(Not run)
```

---

generate_segmentized_network   *Generate segmentized network from road network edges*

Description

The function generates a segmentized network in the specified area. A segmentized network is a network in which the edges of a road network are divided into segments of a specified length. The function creates two tables representing edges and nodes of the network.

The function generates a segmentized network by the following steps:

1. Split road network edges into segments as segmentized network edges
2. Generate segmentized network nodes from segmentized network edges
3. Store the segmentized network edges and nodes into the database

The node and edge tables of the segmentized network have the same columns as the road network node and edge tables.

Usage

```
generate_segmentized_network(
  conn,
  source_table = "road_edges",
  node_table = "segment_nodes",
  edge_table = "segment_edges",
  segment_length = 10
)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.
`source_table`	A `character`, the name of the road network edge table (default: `road_edges`).
`node_table`	A `character`, the name of the segmentized network node table (default: `segment_nodes`).
`edge_table`	A `character`, the name of the segmentized network edge table (default: `segment_edges`).
`segment_length`	A `numeric`, the length of segments in meters (default: 10).

Value

Number of rows of data updated.

See Also

generate_road_network

Examples

```
## Not run:
generate_segmentized_network(con)

## End(Not run)
```

---

launch_database *Launch the database*

Description

Launch the database using docker. This function requires docker to be installed beforehand.

Usage

```
launch_database(
  user = "admin",
  password = "p@ssw0rd",
  dbname = "roadnetdb",
  port = 5432
)
```

Arguments

user	The user name (default: "admin").
password	The password (default: "p@ssword").
dbname	The database name (default: "roadnetdb").
port	The port number (default: 5432).

Value

The return code of this process.

Examples

```
## Not run:
launch_database()

## End(Not run)
```

`register_osm_roads` *Register OpenStreetMap road data into the database*

Description

Register OpenStreetMap road data into the database. If the specified road data has already been registered, this function update the data by specified road data as new data.

Usage

```
register_osm_roads(conn, osm_roads)
```

Arguments

`conn`	A `DBIConnection`, a connection object to the database.
`osm_roads`	A data frame of OpenStreetMap road data.

Value

Number of rows of data updated.

Examples

```
## Not run:
toyota_roads <- read_osm_roads(toyota)
register_osm_roads(con, toyota_roads)

## End(Not run)
```

`shutdown_database` *Shutdown the database*

Description

Shutdown the database using docker. This function requires docker to be installed beforehand.

Usage

```
shutdown_database()
```

Value

The return code of this process.

Examples

```
## Not run:
shutdown_database()

## End(Not run)
```

Index