



Network Interdependency Database Schema Documentation

Overview

This document details the functions that exist within the base network_interdependency database, including any trigger functions that execute on specific tables. The document separates the callable functions from these auto-operating trigger functions.

Callable Functions

Function Name: ni_reset_database

Function Description: Resets the network interdependency database by removing all tables other than Nodes, Edges, Edge_Geometry, Graphs and Global Interdependency, as well as resetting the serial keys for Graphs and Global Interdependency. Finally all records from the geometry_columns table are also deleted.

Parameters in:

- Schema_name: string

Returns:

- Void

Function Name: ni_delete_network

Function Description: Deletes all tables associated with the network prefix supplied, including interdependencies as well as removing associated records from the Graphs and Global Interdependency tables.

Parameters in:

- Network_name_prefix: string

Returns:

- Void

Function Name: ni_check_srid

Function Description: Checks against the PostGIS spatial_ref_sys table that the supplied EPSG number e.g. 27700 exists i.e. is a valid SRID value.



Parameters in:

- SRID: integer

Returns:

- Boolean

Function Name: ni_create_network_tables

Function Description: Function to create empty inherited Nodes, Edges and Edge_Geometry tables for a particular network, based on the table prefix and SRID values supplied e.g. if the table prefix is **transport**, then **transport_Nodes**, **transport_Edges** and **transport_Edge_Geometry** tables will be created. The function calls the individual functions ni_create_network_table_nodes, ni_create_network_table_edge_geometry and ni_create_network_table_edges to create the tables (see later function definitions for more details).

Parameters in:

- Table_prefix: string
- SRID: integer

Returns:

- Boolean

Function Name: ni_create_network_table_nodes

Function Description: Function to create an empty inherited Nodes table, based on the table prefix and SRID supplied e.g. if the table prefix is **transport**, then **transport_Nodes** will be created.

Parameters in:

- Table_prefix: string
- SRID: integer

Returns:

- Boolean

Function Name: ni_create_network_table_edges

Function Description: Function to create an empty inherited Edge table, based on the table prefix supplied e.g. if the table prefix is **transport**, then **transport_Edges** will be created. No SRID is required here because the geometry for the Edge is stored in a separate table Edge_Geometry



Parameters in:

- Table_prefix: string

Returns:

- Boolean

Function Name: ni_create_network_table_edge_geometry

Function Description: Function to create an empty edge_geometry table, based on the table prefix and SRID supplied e.g. if the table prefix is *transport*, then *transport_Edge_Geometry* will be created.

Parameters in:

- Table_prefix: string

Returns:

- Boolean

Function Name: ni_add_geometry_columns

Function Description: Function to add the references to the inherited Nodes and Edge_Geometry table to the PostGIS geometry_columns table, based on the table prefix supplied e.g. if the table prefix is *transport* then *transport_Nodes* and *transport_Edge_Geometry* will be created.

Parameters in:

- Table_prefix: string
- SRID: integer

Returns:

- Boolean

Function Name: ni_add_fr_constraints

Function Description: Function to add the appropriate foreign key constraints to the Nodes, Edges, and Edge_Geometry tables.

- Foreign key from Nodes table to Graphs table (GraphID to GraphID)
- Foreign key from Edges table to Nodes table (Node_F_ID to NodeID)
- Foreign key from Edges table to Nodes table (Node_T_ID to NodeID)
- Foreign key from Edges table to Edge_Geometry table (Edge_GeomID to GeomID)
- Foreign key from Edges table to Graphs table (GraphID to GraphID)

Parameters in:



- Table_prefix: string

Returns:

- Boolean

Comment [s1]: Does this really need to be Boolean, or can we just return void.

Function Name: ni_add_graph_record

Function Description: Function to add a new record to the graphs table based on the table prefix supplied, and Boolean values indicating if the graph is directed or a multigraph.

Parameters in:

- Table_prefix: string
- Directed: Boolean (true if directed, false otherwise)
- MultiGraph: Boolean (true if directed, false otherwise)

Returns:

- Void

Function Name: ni_create_edge_view

Function Description: Function to create a view that joins the Edge and Edge_Geometry tables based on the supplied table prefix. It also sets the appropriate geometry constraints on the view, and adds its table to the PostGIS geometry_columns table.

Comment [s2]: Need to ensure that when deleting the network/resetting the database that any views created are also deleted.

Parameters in:

- Table_prefix: string

Returns:

- New_edge_edge_geometry_view table name: string

Function Name: ni_create_interdependency_edge_view



Function Description: Function to create a view that joins the Interdependency and Interdependency_Edge tables based on the supplied table prefix. It also sets the appropriate geometry constraints on the view, and adds its table to the PostGIS geometry_columns table.

Parameters in:

- Table_prefix: string

Returns:

- New_interdependency and interdependency_edge _view table name: string

Function Name: ni_create_interdependency_tables

Function Description: Function to create the empty interdependency and interdependency_edge tables, based on the supplied table prefixes e.g. if the table prefixes are **transport** and **energy**, then the resulting tables will be **transport_energy_Interdependency** and **transport_energy_Interdependency_Edge**. This function also adds the appropriate foreign key constraints:

- New Interdependency table to Graphs table (Interdependency_Graphs_F_GraphID to GraphID)
- New Interdependency table to Graphs table (Interdependency_Graphs_T_GraphID to GraphID)
- New Interdependency table to the equivalent Nodes table (Interdependency_Nodes_F_NodeID to NodeID)
- New Interdependency table to the equivalent Nodes table (Interdependency_Nodes_T_NodeID to NodeID)

This function also adds a record to the Global_Interdependency tables once they have been created.

Parameters in:

- Table_prefix_network_1: string
- Table_prefix_network_2: string

Returns:

- Boolean

Function Name: ni_edge_geometry_equality_check



Function Description: A function to determine if the input edge geometry matches an edge geometry that already exists in the edge geometry table. This exploits the ST_Equals PostGIS function to determine equality.

Parameters in:

- Table prefix: string
- Geometry WKT: string
- SRID: integer

Returns:

- Matched geometry id: integer (-1 if not matched)

Function Name: `ni_node_geometry_equality_check`

Function Description: A function to determine if the input node geometry matches a node that already exists in the Node table. This exploits the ST_Equals PostGIS function to determine equality.

Parameters in:

- Table prefix: string
- Geometry WKT: string
- SRID: integer

Returns:

- Matched geometry id: integer (-1 if not matched)

Returns:

Function Name: `ni_check_network_tables`

Function Description: A function that checks whether the 3 necessary tables forming a network or graph within the database schema are present i.e. `_Nodes`, `_Edges`, `_Edge_Geometry`.

Parameters in:

- Table prefix: string

Returns:

- Boolean

Function Name: `ni_check_interdependency_tables`



Function Description: A function that checks whether the 2 necessary tables exist that form a valid interdependency within the database schema i.e. `_Interdependency` and `_Interdependency_Edge`.

Parameters in:

- Table prefix network 1: string
- Table prefix network 2: string

Returns:

- Boolean

Function Name: `ni_graph_to_csv`

Function Description: A function to return all tables related to the specified input table prefix as .csv files, including all interdependency tables and views

Parameters in:

- Table prefix: string
- CSV File prefix: string
- CSV File path: string

Returns:

- Void



Trigger Functions

Function Name: check_record_geometry_columns_table_post_graph_insert

Function Description: A function to check that the values for Nodes and Edges in the newly added Graphs record, are tables that have been added to the geometry_columns table. If the tables have not been added, this trigger will add the reference.

Acts on: Graphs

When: After Insert

Row / Statement: Row

Function Name: check_record_geometry_columns_table_post_graph_nodes_update

Function Description: A function to check that the newly updated value for the Nodes column of the Graphs table i.e. name of a Node table, has been added to the geometry_columns table. If the table has not been added, this trigger will add the reference.

Acts on: Nodes of Graphs

When: After Update

Row / Statement: Row

Function Name: check_record_geometry_columns_table_post_graph_edges_update

Function Description: A function to check that the corresponding Edge_Geometry table to the newly updated value for the Edges column of the Graphs table, has been added to the geometry_columns table. If the table has not been added, this trigger will add the reference to the Edge_Geometry table.

Acts on: Edges of Graphs

When: After Update

Row / Statement: Row



Function Name: delete_record_geometry_columns_table_post_delete_graph

Function Description: A function to delete the tables in the geometry_columns table once a record has been deleted from the Graphs table i.e. will remove any Node, Edge_Geometry references

Acts on: Graphs

When: After Delete

Row / Statement: Row

Function Name: delete_int_tables_post_delete_graph

Function Description: A function to remove the interdependency, interdependency_edge tables and any related views when a record is deleted from the Graphs table.

Acts on: Graphs

When: After Delete

Row / Statement: Row

Function Name: delete_nodes_table_post_delete_graph

Function Description: A function to delete the Nodes table, based on the recent value stored in the Nodes column of the recently deleted Graph record.

Acts on: Graphs

When: After Delete

Row / Statement: Row

Function Name: delete_edges_table_post_delete_graph

Function Description: A function to delete the Edges table, based on the recent value stored in the Edges column of the recently deleted Graph record.

Acts on: Graphs

When: After Delete

Row / Statement: Row



Function Name: delete_edges_geometry_table_post_delete_graph

Function Description: A function to delete the Edge_Geometry table, based on the recent value stored in the Edges column of the recently deleted Graph record.

Acts on: Graphs

When: After Delete

Row / Statement: Row

Function Name: delete_int_table_post_int_record_delete

Function Description: A function to delete the corresponding interdependency table based on the recent value stored in InterdependencyTableName of the deleted record from Global_Interdependency

Acts on: Global_Interdependency

When: After Delete

Row / Statement: Row

Function Name: delete_global_int_record_post_graph_record_delete

Function Description: A function to delete the records from the Global_Interdependency table equivalent to the recently deleted graphs deleted from the Graphs table.

Acts on: Graphs

When: After Delete

Row / Statement: Row