**ni\_data\_proc\_connect\_hanging\_edges\_to\_node\_like** – this function connects always **edge “ends”** to each node that are related by some relationship between two attributes; one attribute in the nodes table, and one attribute in the edges table.

For example an attribute called “site\_name” in the nodes table may relate, ***in some way***, to an attribute called “connects” in the edges table e.g.

|  |  |
| --- | --- |
| ID | Site\_name |
| 1 | Scunthorpe |
| 2 | Barrow |
| 3 | Hull |

Table.1 – Example node table

|  |  |
| --- | --- |
| ID | Connects |
| 10 | Scunthorpe to Immingham |
| 11 | Carlisle to Barrow |
| 12 | Hull to York |

Table.2 – Example edge table

Here we can see that by searching the values of the “connects” attribute of the edges table using the values of the “site\_name” attribute, we will be able to select only those edges that have the appropriate “site\_name” value as their “connects” attribute. So when we encounter “Scunthorpe” in the nodes “site\_name” attribute, the function will select all links with a “connects” attribute that contains “Scunthorpe”. At this point an additional link is created between the node “Scunthorpe” and the edge “Scunthorpe to Immingham”.

**Parameters:**

1) Edge\_table\_name: string e.g. data\_national\_grid\_gas\_pipeline\_feeder

2) Edge\_Geometry table geometry column name: string e.g. geom.

3) Edge table primary key: string e.g. gid

4) Edge table attribute: string - relates in some way to the node table attribute defined in parameter 8)

5) Node\_table\_name: string e.g. data\_national\_grid\_gas\_site\_centroids

6) Node table geometry column name: string e.g. geom.

7) Node table primary key: string e.g. gid

8) Node table attribute: string - relates in some way to the node edge table attribute defined in parameter 4)

9) Relationship proxy: integer – this maps to a number of very simple relationships that define how the edge table attribute (4) relates to the node table attribute (8).

|  |  |
| --- | --- |
| Proxy | Description |
| 0 | Equality i.e. WHERE edge\_table\_attribute = node\_table\_attribute, create a new link |
| 1 | Case Insensitive comparison (no wildcards) e.g. edge\_table\_attribute ILIKE node\_table\_attribute |
| 2 | Case Sensitive comparison (no wildcards) e.g. edge\_table\_attribute LIKE node\_table\_attribute |
| 3 | Case Insensitive comparison (wildcards) e.g. edge\_table\_attribute ILIKE %node\_table\_attribute% OR edge\_table\_attribute ILIKE %node\_table\_attribute OR ILIKE edge\_table\_attribute ILIKE node\_table\_attribute% |
| 4 | Case Sensitive comparison (wildcards) e.g. edge\_table\_attribute LIKE %node\_table\_attribute% OR edge\_table\_attribute LIKE %node\_table\_attribute OR ILIKE edge\_table\_attribute LIKE node\_table\_attribute% |
| 5 | Case Insensitive comparison where edge\_table\_attribute is usually suffix of node\_table\_attribute e.g. edge\_table\_attribute ILIKE %node\_table\_attribute |
| 6 | Case Sensitive comparison where edge\_table\_attribute is usually suffix of node\_table\_attribute e.g. edge\_table\_attribute LIKE %node\_table\_attribute |
| 7 | Case Insensitive comparison where edge\_table\_attribute is usually prefix of node\_table\_attribute e.g. edge\_table\_attribute ILIKE node\_table\_attribute% |
| 8 | Case Sensitive comparison where edge\_table\_attribute is usually prefix of node\_table\_attribute e.g. edge\_table\_attribute LIKE node\_table\_attribute% |

e.g. **SELECT \* FROM ni\_data\_proc\_connect\_hanging\_edges\_to\_node\_like ('data\_national\_grid\_gas\_pipeline\_feeder','geom','gid','inspecti\_1','data\_national\_grid\_gas\_site\_centroids','geom','gid','derived\_site\_name', 0, 'testing0', false) f(gid integer, objectid numeric, inspecti\_1 varchar(254), subtypecd numeric(10,0), primaryind numeric(10, 0), gavprimary numeric(10, 0), pipe\_name varchar(255), ng\_owned varchar(2), geom geometry, connection\_point\_geom geometry, additional\_geom geometry, additional\_combined\_geom geometry, start\_point\_distance numeric(10,0), end\_point\_distance numeric(10,0)) ORDER BY gid ASC;**

This table would be output with \_join appended to the input output table name (parameter 7). Secondly a table with \_unique appended to the input output table name (parameter 7) is also written to the schema that contains the original geometry replaced with the newly derived geometry.