

* Import a Shapefile into Oracle Spatial as a normal vector map; * Replace all occurrences of SHP_TABLE (case sensitive, can be part of a word) in the script below with the table name of the imported map from Shapefile (copy all script from below to a text editor and perform that replace); * Copy/Paste (in Linux: select text and then use middle mouse button or Shift+Insert to paste it in a console) the script line-by-line, follow any severe errors (ignore ones, when it writes that table cannot be dropped when it does not actually exist :) and the like).

```
-- delete any tables existing
EXECUTE SDO_NET.DROP_NETWORK('SHP_TABLE_NET');
DROP TABLE SHP_TABLE_PLINK$;
DROP TABLE SHP_TABLE_EDGE$;
DROP TABLE SHP_TABLE_NODE$;
DROP TABLE SHP_TABLE_LINK$;
DROP TABLE SHP_TABLE_FACE$;

-- delete any existing geometry layer
EXECUTE SDO_TOPO.DELETE_TOPO_GEOMETRY_LAYER('SHP_TABLE', 'SHP_TABLE_TOPO',
'FEATURE');
DROP TABLE SHP_TABLE_TOPO;

-- then drop existing topology
EXECUTE SDO_TOPO.DROP_TOPOLOGY('SHP_TABLE');
-- create new topology
EXECUTE SDO_TOPO.CREATE_TOPOLOGY('SHP_TABLE', 0.5);
-- insert the universal face for an empty topology
INSERT INTO SHP_TABLE_FACE$(face_id, boundary_edge_id, island_edge_id_list,
island_node_id_list, mbr_geometry)
VALUES(-1, NULL, SDO_LIST_TYPE(), SDO_LIST_TYPE(), NULL);
-- create an Oracle table with a feature layer
DROP TABLE SHP_TABLE_TOPO;
-- note, do not create unique id (PRIMARY KEY) as the data in teleatlas.dbf
is inconsistent
CREATE TABLE SHP_TABLE_TOPO(id NUMBER, feature SDO_TOPO_GEOMETRY);
--CREATE TABLE SHP_TABLE_TOPO(id number, type varchar2(1), feature
SDO_TOPO_GEOMETRY);

-- register feature layer with topology
EXECUTE SDO_TOPO.ADD_TOPO_GEOMETRY_LAYER('SHP_TABLE', 'SHP_TABLE_TOPO',
'FEATURE', 'CURVE');
-- create updatable TOPO_MAP object and load the whole topology in cache
-- since we just created the topology, the cache will be empty
EXECUTE SDO_TOPO_MAP.DROP_TOPO_MAP('SHP_TABLE_MAP_CACHE');
EXECUTE SDO_TOPO_MAP.CREATE_TOPO_MAP('SHP_TABLE', 'SHP_TABLE_MAP_CACHE');
EXECUTE SDO_TOPO_MAP.LOAD_TOPO_MAP('SHP_TABLE_MAP_CACHE', 'true');

-- copy/paste everything in one go between BEGIN and COMMIT; (including):
-- this one will demand a considerable amount of time
BEGIN
FOR r IN (SELECT id, geometry FROM SHP_TABLE) LOOP
-- associate topological primitives with features
INSERT INTO SHP_TABLE_TOPO(id, feature)
```

```

VALUES(r.id,
      SDO_TOPO_MAP.CREATE_FEATURE('SHP_TABLE',
                                   'SHP_TABLE_TOPO',
                                   'FEATURE',
                                   r.geometry)
      );

END LOOP;
END;
COMMIT;

-- commit topology changes
EXECUTE SDO_TOPO_MAP.COMMIT_TOPO_MAP;
EXECUTE SDO_TOPO_MAP.DROP_TOPO_MAP('SHP_TABLE_MAP_CACHE');

-- check how many primitives were converted (should be the same number as
primitives in the DB)
SELECT COUNT(*) FROM SHP_TABLE_TOPO;

-- after an initial bulk load into an empty topology, initialize_metadata
EXECUTE SDO_TOPO.INITIALIZE_METADATA('SHP_TABLE');

-- check here what tables have been created:
SELECT table_name FROM user_tables WHERE table_name LIKE 'SHP_TABLE_%$';

CREATE TABLE SHP_TABLE_LINK$
AS SELECT
      edge_id AS link_id,
      start_node_id,
      end_node_id,
      geometry
FROM SHP_TABLE_EDGE$;

-- FULL NETWORK:
--CREATE TABLE SHP_TABLE_PATH$(PATH_ID NUMBER,
--      PATH_NAME          VARCHAR2(200),
--      PATH_TYPE          VARCHAR2(200),
--      START_NODE_ID      NUMBER NOT NULL,
--      END_NODE_ID        NUMBER NOT NULL,
--      COST                NUMBER,
--      SIMPLE              VARCHAR2(1),
--      PATH_GEOMETRY       MDSYS.SDO_GEOMETRY);
--CREATE TABLE roads_incompl_plink$(path_id number not null, link_id number
not null, seq_no number not null);
--INSERT INTO USER_SDO_NETWORK_METADATA(
--      network, network_category, geometry_type,
--      node_table_name, node_geom_column,
--      link_table_name, link_geom_column, link_direction,
--      path_table_name, path_geom_column,
--      path_link_table_name)
--      VALUES(
--      'SHP_TABLE_NET', 'SPATIAL', 'SDO_GEOMETRY',
--      'SHP_TABLE_NODE$', 'GEOMETRY',

```

```
-- 'SHP_TABLE_LINK$', 'GEOMETRY', 'UNDIRECTED',
-- 'SHP_TABLE_PATH$', 'PATH_GEOMETRY',
-- 'SHP_TABLE_PLINK$');

-- MINIMAL NETWORK:
INSERT INTO USER_SDO_NETWORK_METADATA(
    network, network_category, geometry_type,
    node_table_name, node_geom_column,
    link_table_name, link_geom_column, link_direction)
VALUES(
    'SHP_TABLE_NET', 'SPATIAL', 'SDO_GEOMETRY',
    'SHP_TABLE_NODE$', 'GEOMETRY',
    'SHP_TABLE_LINK$', 'GEOMETRY', 'UNDIRECTED');

SELECT SDO_NET.VALIDATE_NETWORK('SHP_TABLE_NET') FROM DUAL;

COMMIT;
```

If anything, contact me (the creator of this page) {at} unibz {dot} it.

From:

<https://wiki.inf.unibz.it/> - **Engineering-Tech Wiki**

Permanent link:

<https://wiki.inf.unibz.it/doku.php?id=contrib:shp2ndm&rev=1184945081>

Last update: **2019/01/16 10:03**

