SOFTWARE. HARDWARE. COMPLETE.


ORACLE®

How to Build Network Applications with Oracle Spatial Network Data Model

Daniel Geringer
Senior Software Development Manager
Oracle's Spatial Technologies

Presentation Outline

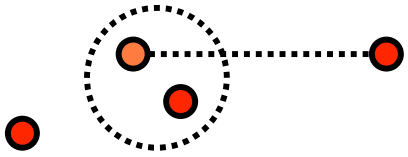
- Overview of the Oracle Spatial Network Data Model (NDM)
- New NDM tutorial / demo
  - Integrates the Oracle Spatial Network Data Model, Geocoder and Oracle MapViewer
  - Source code examples (similar to MapViewer demo)
  - Download from http://spatial.samplecode.oracle.com under the NDM tab
- How the Oracle Spatial Router leverages NDM

Spatial Analysis Versus Network Analysis

- Oracle Locator and Oracle Spatial solve spatial proximity problems.
- Another type of analysis that is required by users and applications is network analysis.
- Network applications deal with the connectivity of features. Spatial data is optional.

Spatial Closest feature

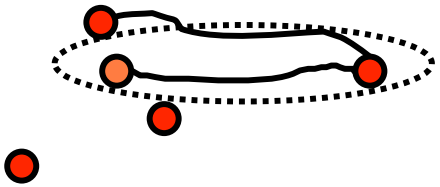
(based on distance)



Verses

Network Closest feature

(based on connectivity and cost)



What Is the Oracle Spatial Network Data Model?

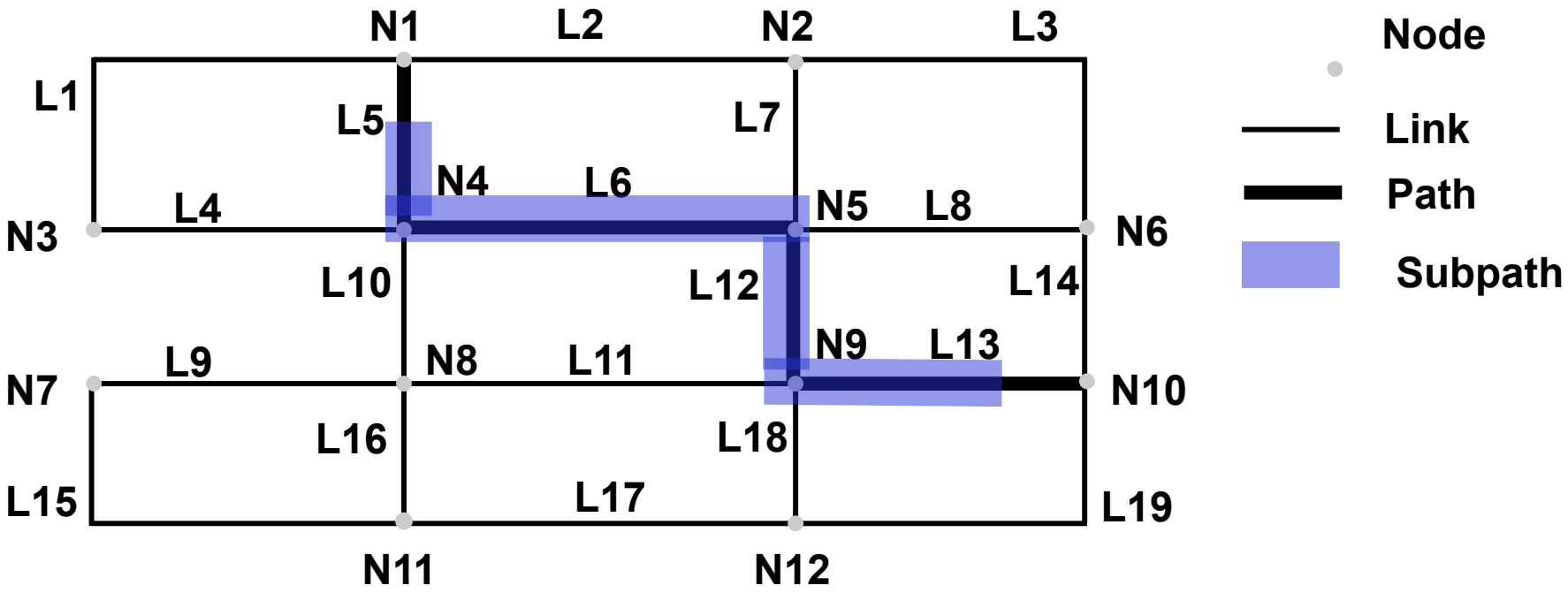
- An open data data model to store and analyze network data.
- Connectivity is determined using nodes and links:
  - Each link has a start node and an end node.
  - Links and/or nodes can have costs
  - Links can be one way or bi-directed

What Is the Oracle Spatial Network Data Model?

- Analysis is based on connectivity and optionally cost information.

- Network analyses includes:
  - Shortest path analysis
  - Nearest neighbor analysis
  - Within cost analysis
  - Minimum cost spanning tree
  - Traveling salesman problem
  - Reachable/Reaching nodes
  - K-shortest paths analysis

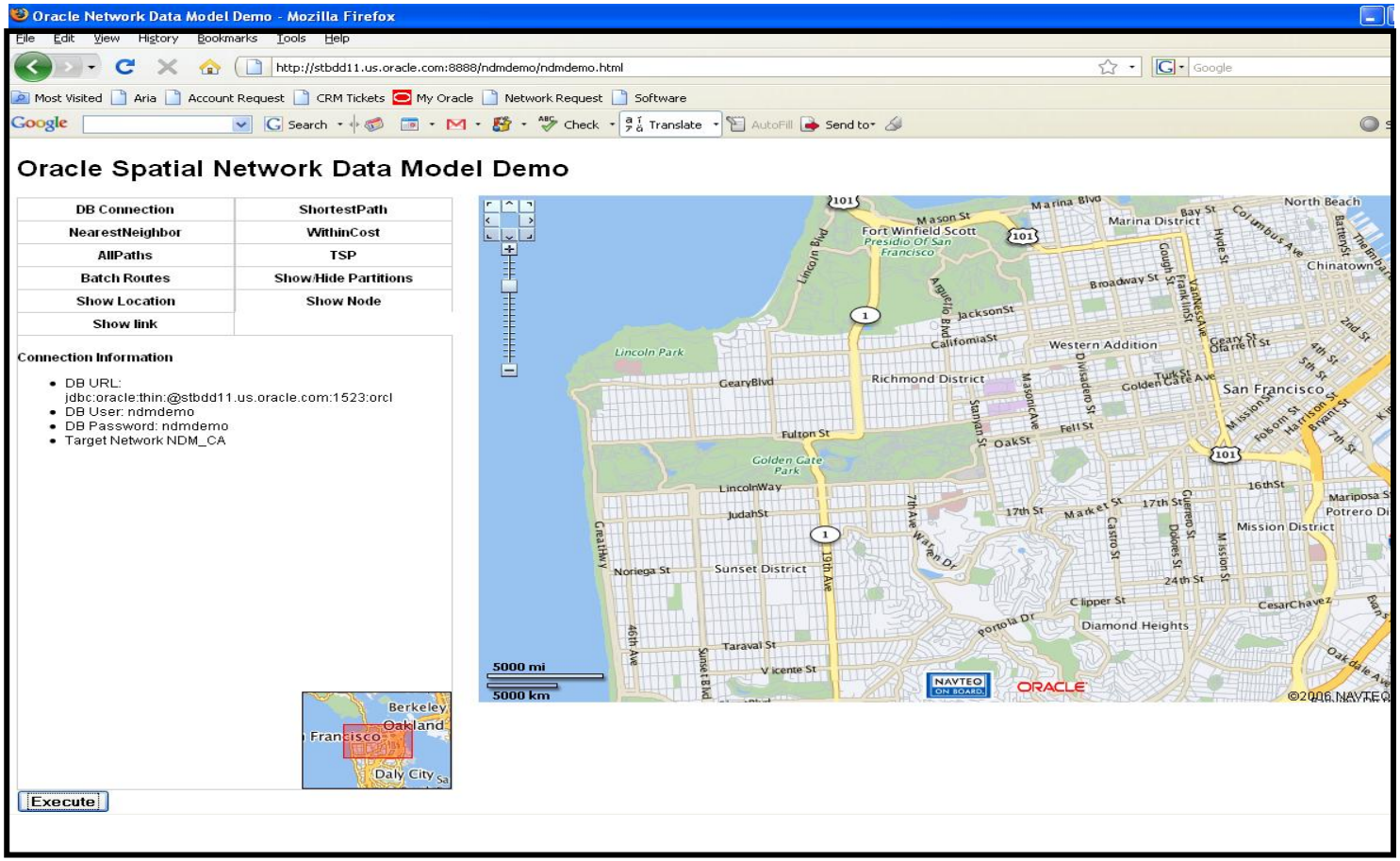
A Simple Road Network



- Nodes are intersections
- Links are streets
- A path is a route between two nodes
- A subpath can represent a route between two addresses

A More Complicated Network

- San Francisco streets



The Node Table (or View)

- Optional columns marked in red.
- Additional user data columns can be added as needed.

```
SQL> desc MYNETWORK_NODE$;
 Name                          Type
 ----------------------------  -------------
 NODE_ID                       NUMBER (Primary Key)
 COST                          NUMBER
 ACTIVE                        VARCHAR2(1)
 GEOMETRY                      SDO_GEOMETRY
```

The Link Table (or View)

- Optional columns marked in red
- Additional user data columns can be added
- Links can be one way or bi-directed
- If bi-directed with a different cost in either direction, add another link.

```
SQL> desc MYNETWORK_LINK$;

 Name                          Type
 ----------------------        -------------
 LINK_ID                       NUMBER (Primary Key)
 START_NODE_ID                 NUMBER
 END_NODE_ID                   NUMBER
 ACTIVE                        VARCHAR2(1)
 LINK_LEVEL                    NUMBER
 COST                          NUMBER
 GEOMETRY                      MDSYS.SDO_GEOMETRY
```

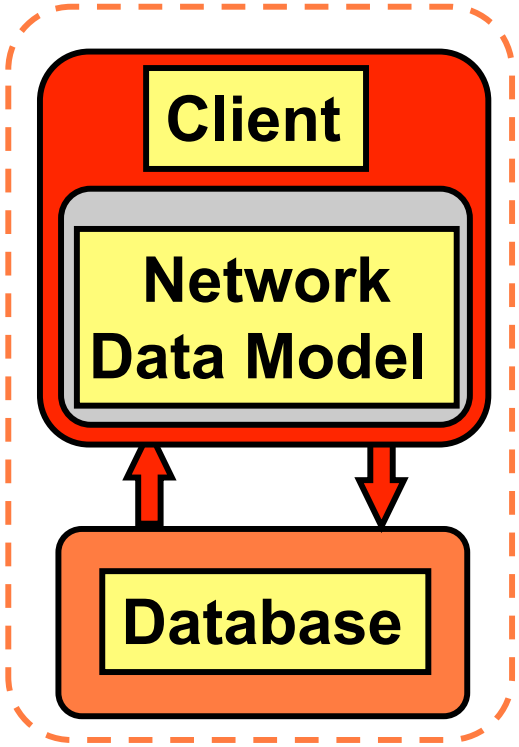
Oracle Spatial Network Data Model API's

- A PL/SQL API to manage the network:
  - Creating network structures, or you can create them manually
  - Validation
- A Java API to perform network analyses:
  - Shortest path
  - within cost,
  - nearest neighbor,
  - k-shortest paths
  - Traveling salesman problem,
  - minimum cost spanning tree
  - Reaching/reachable nodes and others

Oracle Spatial Network Data Model API's

- Oracle Spatial's NDM was introduced in Oracle 10*g*.
- The entire network had to be loaded into memory before any network analysis (not a scalable solution).
- In Oracle 11*g*, load on demand (LOD) was introduced.
- With LOD, the network is divided into partitions. During network analysis, only the relevant partitions are loaded into memory (a very scalable solution).
- The Oracle 10*g* APIs for network analysis have been deprecated in Oracle 11*g*.
- In Oracle 11*g* and later, only use LOD.

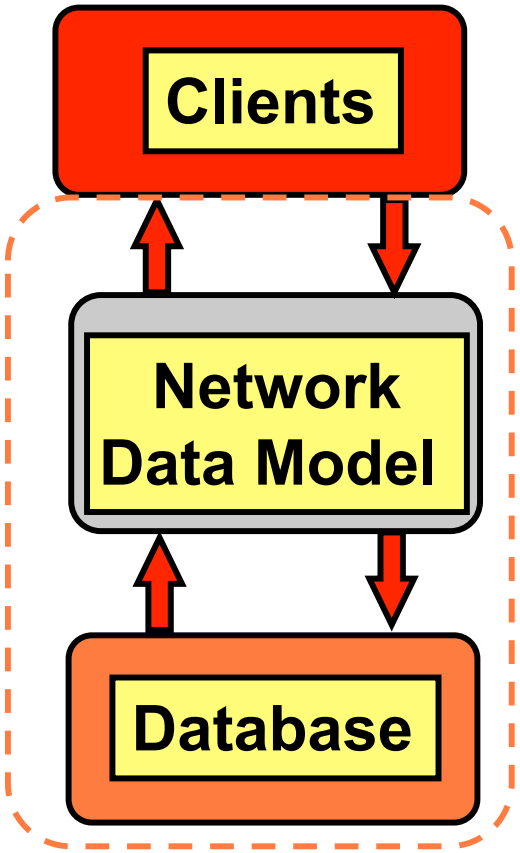
Two-Tier Architecture for NDM



Queries and GUI
Network Loading/Caching/Analysis with the Java API

Persistent Network Data
(Management with PL/SQL API)

Three-Tier Architecture for NDM



Queries and GUI (Web)

Network Loading/Caching/ Analysis with Java API

Persistent Network Data

Required and Optional Persistent Data

- Required persistent data includes:
  - Network metadata
  - Node table (discussed earlier)
  - Link table (discussed earlier)
- Optionally, persistent data can include:
  - Partition table
  - Partition `BLOB` table
  - `SDO_GEOMETRY` representation of nodes and links
  - User data information
  - Path table, path-link table, or subpath table to persist network analysis result

Network Partition Table for Load on Demand (Optional)

```
SQL> desc MYNETWORK_PART$;
 Name                          Type
 ----------------------       -------------
 NODE_ID                       NUMBER
 LINK_LEVEL                    NUMBER
 PARTITION_ID                  NUMBER
```

- If no network partition table, entire network is loaded into memory as a single partition.
- One row for each node.
- Nodes repeated for each hierarchy they belong to (`LINK_LEVEL`).
- You can manually populate a network partition table.
- `SDO_NET.SPATIAL_PARTITION`, partitions a spatial network for you (creates the partition table).

Network Partitioning Best Practices

- Increase the size of the network partition cache. After the cache is full, a Least Recently Used (LRU) algorithm purges partitions from cache to load others.
- Increase Java heap size (Java –Xmx –xms)
- Generate partition `BLOB`s, a binary representation of network partitions that load faster into memory.
- Choose an optimal partition size. Very small partitions may result in excessive partition loads. If partitions are too large, for a network analysis, more data may get loaded than necessary.
- Try to minimize the number of links between partitions.
- Leverage hierarchical networks when possible.

Partition BLOB Table (Optional)

```
SQL> desc MYNETWORK_PBLOB$;
 Name                          Type
 ----------------------       -------------
 LINK_LEVEL                    NUMBER
 PARTITION_ID                  NUMBER
 BLOB                          BLOB
 NUM_INODES                    NUMBER
 NUM_ENODES                    NUMBER
 NUM_ILINKS                    NUMBER
 NUM_ELINKS                    NUMBER
 NUM_INLINKS                   NUMBER
 NUM_OUTLINKS                  NUMBER
 USER_DATA_INCLUDED            VARCHAR2(1)
```

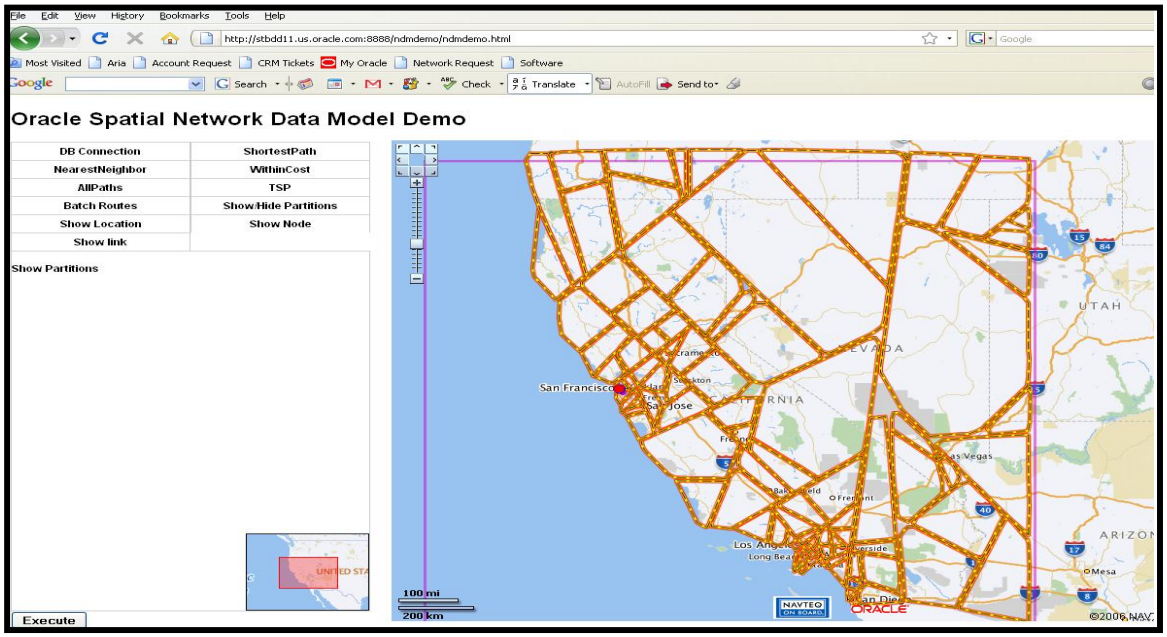
- Partition BLOBs store network partitions in a binary format.
- The binary format is much faster to load into memory.
- Partition BLOBs can't be edited. When changes are made to NDM relational tables, partition BLOBs must be regenerated.
- USER_DATA_INCLUDED is discussed in the next slide.

Partition BLOB Table USER_DATA_INCLUDED (Y or N)

- User data can be considered in a network analysis.
- For example, user data may include speed limits
- User data can be stored in partition BLOBs, or separately in it's own persistent BLOB structure.
  - Advantage - Performance.
  - Disadvantage - Edits to user data requires regeneration of partition BLOBs, including connectivity information that did not change.
- Store static (or fairly static) user data in partition BLOBs. For example, speed limit may be a good candidate in a fairly static road network.
- You can do both, store some user data in partition BLOBs and some outside of the partition BLOBs.

Visualizing Partition Boundaries

- Observe smaller partition boundaries in denser portions of the network (urban areas).
- The number of nodes in each partition is still balanced.



US Network Example

- A US road network with the following characteristics was modeled:
  - 20 million nodes
  - 50 million links
  - 4,096 network partitions
  - Partition size: Approximately 5,500 nodes
  - Node user data: x, y (float)
  - Link user data:
    - Speed limit (float)
    - Road function class (int)
    - For some links, prohibited turn (long)

US Network Example Memory Consumption

- Memory consumption was measured by loading partition `BLOB`s into memory and observing the Java heap size.
- Memory consumption for each partition without user data (only nodes and links) consumed approximately 2 MB.
- Memory consumption for each partition with user data consumed approximately 3 MB.

Optional Tables to Persist Network Analysis Results (Paths)

```
SQL> desc MYNETWORK_PATH$;
 Name                          Type
 ----------------------        -------------
 PATH_ID                       NUMBER (Primary Key)
 PATH_NAME                     VARCHAR2(32)
 PATH_TYPE                     VARCHAR2(24)
 START_NODE_ID                 NUMBER
 END_NODE_ID                   NUMBER
 GEOMETRY                      MDSYS.SDO_GEOMETRY
 COST                          NUMBER
 SIMPLE                        VARCHAR2(1)

SQL> desc MYNETWORK_PLINK$;
PATH_ID                       NUMBER
LINK_ID                       NUMBER
SEQ_NO                        NUMBER
```

Optional Tables to Persist Network Analysis Results (Subpaths)

```
SQL> desc MYNETWORK_SPATH$;
 Name                                      Type
 ----------------------------------------- ----------------
 SUBPATH_ID                                NUMBER
 SUBPATH_NAME                              VARCHAR2(200)
 SUBPATH_TYPE                              VARCHAR2(200)
 REFERENCE_PATH_ID                         NUMBER
 START_LINK_INDEX                          NUMBER
 END_LINK_INDEX                            NUMBER
 START_PERCENTAGE                          NUMBER
 END_PERCENTAGE                            NUMBER
 COST                                      NUMBER
 GEOMETRY                                  SDO_GEOMETRY
```

Network Metadata
Enter One Row Per Network

```
SQL> desc user_sdo_network_metadata;
 Name                           Type
 ------------------------------ -------------
 NETWORK                        VARCHAR2(24)
 NETWORK_ID                     NUMBER
 NETWORK_CATEGORY               VARCHAR2(12)
 GEOMETRY_TYPE                  VARCHAR2(24)
 NETWORK_TYPE                   VARCHAR2(24)
 NO_OF_PARTITIONS               NUMBER
 NODE_TABLE_NAME                VARCHAR2(32)
 NODE_GEOM_COLUMN               VARCHAR2(32)
 NODE_COST_COLUMN               VARCHAR2(32)
 LINK_TABLE_NAME                VARCHAR2(32)
 LINK_GEOM_COLUMN               VARCHAR2(32)
 LINK_DIRECTION                 VARCHAR2(12)
 LINK_COST_COLUMN               VARCHAR2(32)
 PATH_TABLE_NAME                VARCHAR2(32)
 PATH_GEOM_COLUMN               VARCHAR2(32)
 PATH_LINK_TABLE_NAME           VARCHAR2(32)
 PARTITION_TABLE_NAME           VARCHAR2(32)
 PARTITION_BLOB_TABLE_NAME      VARCHAR2(32)
```

Network Data Model Process

- Set up the network data model:
  - Create the node table and the link table.
  - Insert metadata into `USER_SDO_NETWORK_METADATA`.
- Load network and user data:
  - Use SQL/Loader, insert statements, or programmatically
- Optionally, partition the network:
  - Node-based partitions
  - Optional partition `BLOB` generation
- Analyze and/or edit network data:
  - Application
  - Java program

Setting Up the Network Data Model

- Create tables manually:
  - Create the node, link, and other network tables.
  - Table schema is described earlier in this lesson.
  - Populate `USER_SDO_NETWORK_METADATA`.
- Create tables with routines in the `SDO_NET` package:
  - `SDO_NET.CREATE_NODE_TABLE`
  - `SDO_NET.CREATE_LINK_TABLE`
  - `SDO_NET.CREATE_PARTITION_TABLE`
  - `SDO_NET.CREATE_PARTITION_BLOB_TABLE`
  - `SDO_NET.CREATE_PATH_TABLE`
  - `SDO_NET.CREATE_PATH_LINK_TABLE`
- `SDO_NET.SPATIAL_PARTITION`, a routine to partition a spatial network (creates the network partition table)
- `SDO_NET.GENERATE_PARTITION_BLOBS`, a routine to generate a binary representation of network partitions, (creates the partition `BLOB` table)

Optionally Include Geometry in the Network Data Model

- Network analyses is based on graph connectivity and cost.
- Even in a spatial network, the geometry is not considered in the network analyses.
- Network analysis routines can return the geometry representation of a path or subpath.
- The geometry representation is commonly rendered in a mapping application.
- For spatial networks, set the following in `USER_SDO_NETWORK_METADATA`:
  - `GEOMETRY_TYPE = 'SDO_GEOMETRY'`

Partition the Network

- For load on demand, you can partition a spatial network with the `SDO_NET.SPATIAL_PARTITION` routine.
- You provide the partition table name, and `SDO_NET.SPATIAL_PARTITION` creates and populates the partition table.

```
execute SDO_NET.SPATIAL_PARTITION (
  network               =>'NYC_NET',          -- network name
  partition_table_name  =>'NYC_PART$',
  max_num_nodes         => 5000,              -- per partition
  log_loc               =>'DIRECTORY_PATH',   -- for log file
  log_file              =>'nyc_part.log',     -- log file name
  open_mode             =>'a',                -- a for append
  link_level            => 1);                -- hierarchy level
```

Generate Partition BLOBs

- To speed up loading partitions into memory, generate partition BLOBs.
- You provide the partition BLOB table name, and SDO_NET.GENERATE_PARTITION_BLOBS creates and populates the partition BLOB table.
- Partition BLOBs cannot be edited. They must be regenerated to reflect the network edits.

```
execute SDO_NET.GENERATE_PARTITION_BLOBS(
  network                =>'NYC_NET',              -- network name
  link_level             =>1,                      -- hierarchy level
  partition_blob_table_name => 'NYC_PBLOB$',
  include_user_data      => FALSE,                 -- in partition blobs
  log_loc                =>'DIRECTORY_PATH',       -- for log file
  log_file               =>'nyc_part.log',         -- log file name
  open_mode              =>'a');                   -- a for append
```

USER_SDO_NETWORK_USER_DATA Metadata View

- You can persist additional user-defined application data with a network.
- When partitions are loaded into memory, the corresponding user data is also loaded.
- Insert one row in `USER_SDO_NETWORK_USER_DATA` for each user data instance.

```
SQL> desc USER_SDO_NETWORK_USER_DATA;
 Name                          Type
 ----------------------------  -------------
 NETWORK                       VARCHAR2(32)
 TABLE_TYPE                    VARCHAR2(12)
 DATA_NAME                     VARCHAR2(32)
 DATA_TYPE                     VARCHAR2(12)
 DATA_LENGTH                   NUMBER(38)
 CATEGORY_ID                   NUMBER(38)
```

USER_SDO_NETWORK_USER_DATA Metadata View

- TABLE_TYPE: Specifies the table that persists the user data. **NOTE** There is a different way not discussed here to store/access user_data in other tables by implementing a NDM Java class.
  - NODE
  - LINK
  - PATH
  - SPATH
- DATA_NAME:
  - Corresponds to the column name
  - The column name must be in either the node, link, path or subpath tables.

USER_SDO_NETWORK_USER_DATA Metadata View

- DATA_TYPE of the user data:
  - VARCHAR2
  - NUMBER
  - INTEGER
  - SDO_GEOMETRY
- DATA_LENGTH: Length of VARCHAR2 user data

USER_SDO_NETWORK_USER_DATA Metadata View

- CATEGORY_ID:
  - User data can be grouped into categories.
  - A category corresponds to the group of user data that is required by an application to perform network analysis.
  - User data columns should not participate in multiple categories.
  - User data with CATEGORY_ID zero gets loaded into partition BLOBs. Otherwise, user data is stored in its own persistent BLOB structure.

Register User Data Example

```
-- Register the user data column called TOLL_COST
-- in the link table.

INSERT INTO user_sdo_network_user_data
  (network,
   table_type,
   data_name,
   data_type,
   category_id)
VALUES ('MYNETWORK',
        'LINK',
        'toll_cost',
        'NUMBER',
         1);
```

Network Constraints

- Network analyses can be constrained with custom Java code.
- For example, during a shortest path analysis, you may want to implement turn restrictions at certain intersections.
- The steps are as follows:
  - Implement a `LODNetworkConstraint` interface.
  - Pass a constraint to the network analysis functions.
    - The constraint is checked during analysis.
    - Accept or reject the next link or node combination.
  - Network constraints require you to implement five methods.
    - `isSatisfied (LODAnalysisInfo info)`
    - `isCurrentNodePartiallyExpanded (LODAnalysisInfo info)`
    - `isNextNodePartiallyExpanded (LODAnalysisInfo info)`
    - `getUserDataCategories()`
    - `getNumberOfUserObjects()`

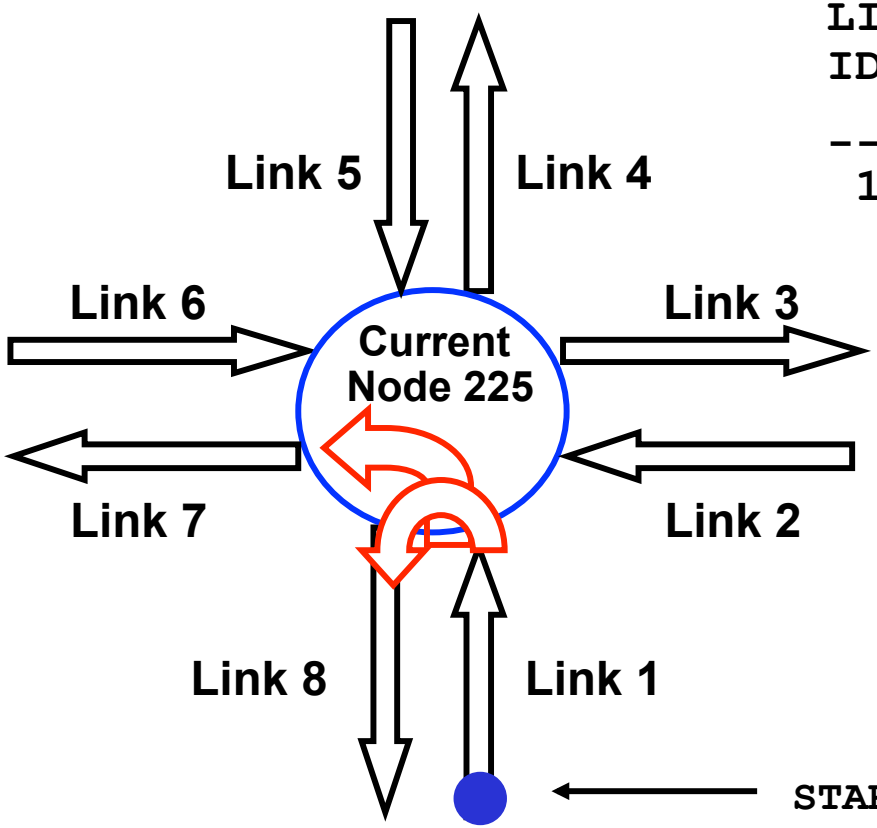
Network Constraint Turn Restriction Example

- Some links have turn restrictions.
- For each link with a turn restriction, store a comma-delimited string of link IDs that cannot follow the link.
- The network constraint implementation parses the comma-delimited string.

```
SQL> desc MYNETWORK_LINK$;

 Name                          Type
 ----------------------        -------------
 LINK_ID                       NUMBER (Primary Key)
 START_NODE_ID                 NUMBER
 END_NODE_ID                   NUMBER
 RESTRICTED_LINKS              VARCHAR2(100)
```

Network Constraint: Turn Restriction Example



LINK ID	START NODE ID	END NODE ID	RESTRICTED LINKS
1	165	225	'7,8'

`START_NODE_ID` for Link 1 = 165

Returning Multiple Costs in a Network Analysis (User-Defined Cost Calculators)

- During analysis, you may want to calculate and report back many costs.
  - For example, the primary cost in a road network may be distance.
  - During a shortest path analysis, you may want to report distance (the primary cost) and toll amount (a secondary cost).
- By default, non primary costs do not influence the network analysis, but they can in a network constraint.

User-Defined Cost Calculators

- Calculate and report back additional costs by writing custom Java classes that implement the LinkCostCalculator or NodeCostCalculator interfaces.
- Steps to calculate a secondary cost for the links in a network analysis:
  - Implement a LinkCostCalculator interface.
  - Pass the link cost calculator to network analysis routine.
  - Multiple cost calculators can be passed to network analysis.
  - Each cost calculator will report back a cost for the resulting path or subpath.

Network Analysis that Considers Traffic Patterns

Routing with Traffic Patterns in NDM

- Fastest Routes are based on speed limits (max. speed)
  - Not very accurate during rush hours
  - A good non-rush route can be the worst rush route!
- Historical average traffic speeds on roads at 15 min, 30 min, 60 min intervals (Navteq Traffic Patterns product)
- Average speeds for:
  - Mon-Thur.
  - Fri.
  - Sat.
  - Sun.
  - holidays
  - special events
- Specify start time in route request with traffic patterns

Routing with Traffic Patterns in NDM – With a Cost Calculator

- Model Traffic Patterns as user data in NDM
  - Generic traffic pattern metadata and schema
  - Generate traffic pattern user data automatically
  - Currently Support Navteq Traffic Patterns
- Link Cost is a time-dependent function that returns travel time
  - Start time determines which set of traffic pattern to use (M-T,Fri.Sat. Sun. Holidays)
  - Link distance/Link speed limit (for non-covered links)
  - Actual Speeds from traffic patterns (interval in a day)
  - Actual Travel Time = Link Distance/Actual Speed

Routing with Traffic Patterns in NDM (Faster Routes at Different Start Times)

Oracle Spatial Network Data Model Demo

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID@percentage, or address.

Start 199488837

End 199919135

Network Constraints
(Hold ctrl key for multi-select or de-select)

custom.NoHighwayConstraint
custom.ProhibitedZoneConstraint
oracle.spatial.router.ndm.TruckHeightConstraint
oracle.spatial.router.ndm.TruckLegalConstraint

Prohibited Zone Draw

Link Cost Calculators
custom.TrafficLinkCostCalculator

Keep Previous Results ☑
Reverse Direction ☐

Include Traffic data ☑
Start Time 10:00 PM

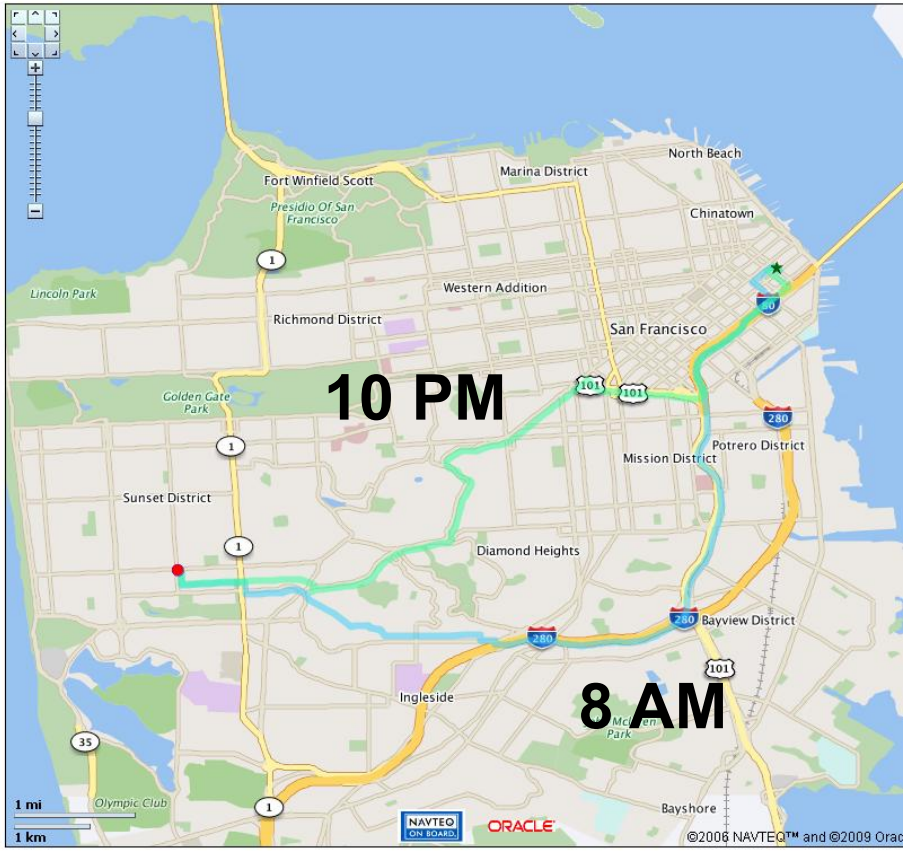
Find Shortest Path

Analysis Result:
(199488837->199919135)
[cost:946.05814, 105 links]

Time to analyze the network: 0.467s.
Time to compute geometries: 0.035s.

Analysis Result:
(199488837->199919135)
[cost:872.93101, 172 links]

Time to analyze the network: 0.436s.
Time to compute geometries: 0.039s.



More Resources for the Network Data Model

- *Oracle Spatial Topology and Network Data Models Developer's Guide*
- Latest javadoc for LOD Java API available:
  - For 11.1.0.6, on Metalink in Oracle 11.1.0.6 LOD Patch 7700528
  - For 11.1.0.7, in `ORACLE_HOME\md\doc\sdonmlod.zip`
- Network data model white paper:
  - http://www.oracle.com/technology/products/spatial
  - *Oracle Database 11g: A Load-On-Demand Approach to Handling Large Networks in the Oracle Spatial Network Data Model*

More Resources for the Network Data Model

- Java examples:
  - Ship with Oracle Database 11*g* Examples media.
  - After Examples media is installed, examples can be found in `ORACLE_HOME\md\demo\network\examples\java\src\lod`
- NDM tutorial/demo
  - Integrates the Oracle Spatial Network Data Model, Geocoder and Oracle MapViewer
  - Source code examples (similar to MapViewer demo)
  - Download from http://spatial.samplecode.oracle.com/ under the NDM tab

Network Data Model
Tutorial / Demo

Oracle Spatial Router Based on NDM in Oracle 11g Release 2

- Reason for this change:
  - Scalability of LOD
  - Flexibility of network constraints
- Oracle Spatial router now supports trucking data from NAVTEQ Transport Product
  - Truck information associated with road networks included in NVT_TRANSPORT table.
  - For example, max truck height, weight, width, and more…
  - Very easy to consider these attributes with network constraints

Truck Routing Request

- <?xml version="1.0" standalone="yes"?>
  <route_request id="8"
  route_preference="shortest"
  road_preference="highway“
  vehicle_type="truck"
  return_driving_directions="true"
  distance_unit="mile"
  time_unit="hour“
  return_route_geometry="false" >
- Route Request vehicle_type
  - (auto|truck) optional, defaults to auto
- Route Request truck_type
  - truck_type: (delivery|public|trailer) optional, no default

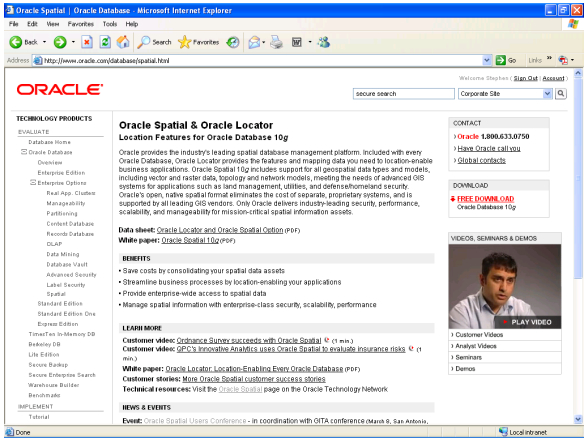
Truck Routing Attributes

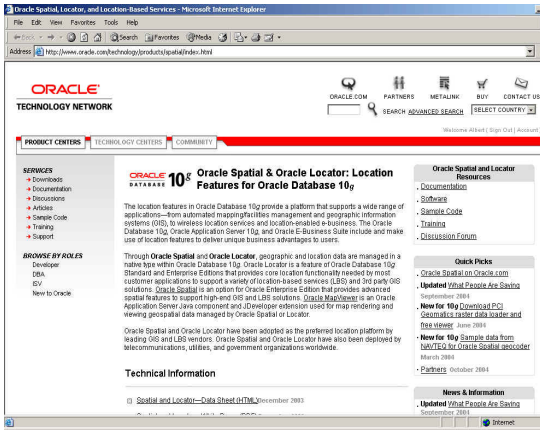
- truck_height: (positive float) optional, no default
- truck_length: (positive float) optional, no default
- truck_per_axle_weight: (positive float) optional, no default
- truck_weight: (positive float) optional, no default
- truck_width: (positive float) optional, no default
- length_unit: (metric|us) optional, default US
- weight_unit: (metric|us) optional, default US
- Truck height, length and width are specified in length_unit units
- Truck per axle weight and weight and specified in weight_unit units

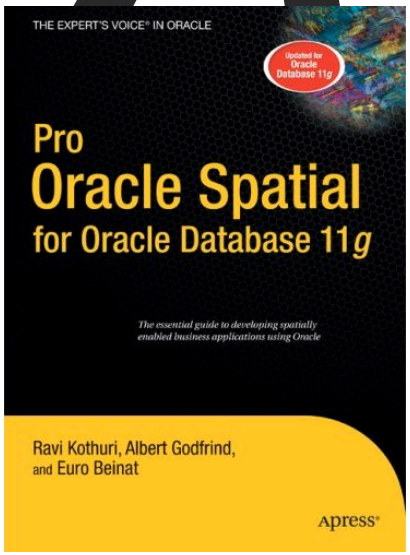
Find out more...

oracle.com/database/spatial.html

oracle.com/technology/products/spatial







oracle.com/technology/products/spatial/htdocs/pro_oracle_spatial.html

SOFTWARE. HARDWARE. COMPLETE.


ORACLE®