Maps, 3D, Tracking, JSON, and Location Analysis: What's New with Oracle Spatial Technologies

Siva Ravada
Senior Director of Development
Spatial and Graph

Safe Harbor Statement for Oracle Database 12.2

The information contained in this document is for informational sharing purposes only, and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

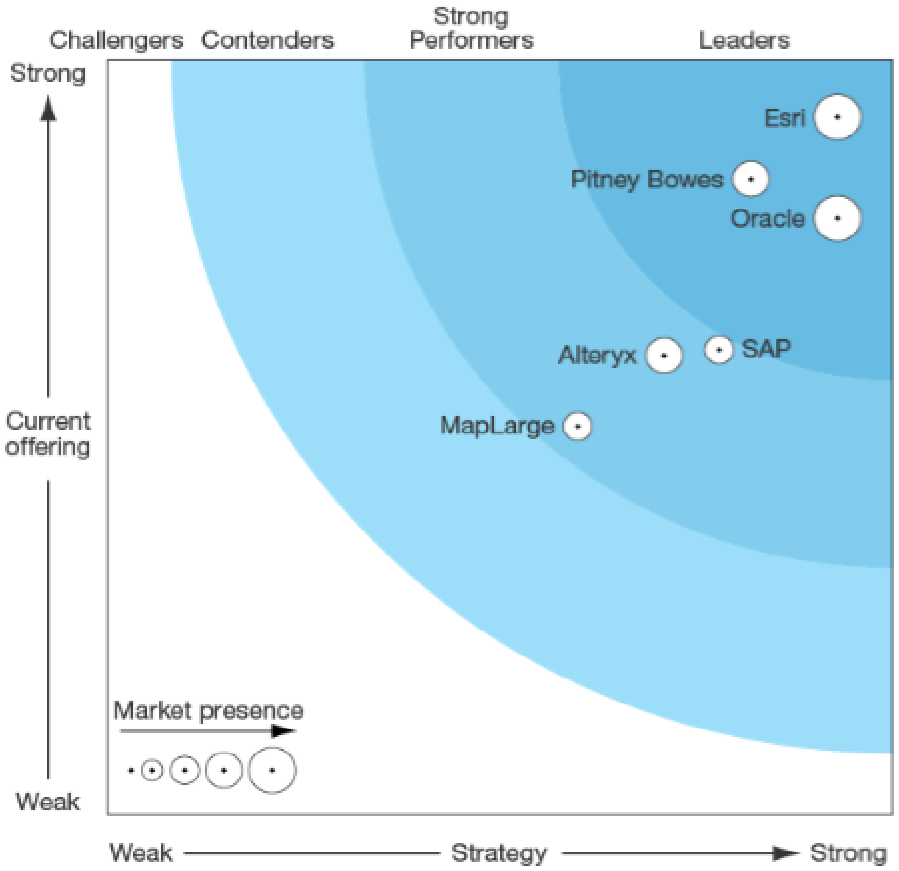
This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

The Forrester Wave™:
Geospatial Analytics Tools And Platforms, Q3 2016

FORRESTER®

WAVE LEADER 2016

Geospatial Analytics Tools And Platforms



The Forrester Wave™ is copyrighted by Forrester Research, Inc. Forrester and Forrester Wave™ are trademarks of Forrester Research, Inc. The Forrester Wave™ is a graphical representation of Forrester's call on a market and is plotted using a detailed spreadsheet with exposed scores, weightings, and comments. Forrester does not endorse any vendor, product, or service depicted in the Forrester Wave. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.

"While hardcore GIS professionals may start their work in other applications, when they want to solve spatial problems in production and with web- and IoT- scale data, Oracle gives them the platform to do so."

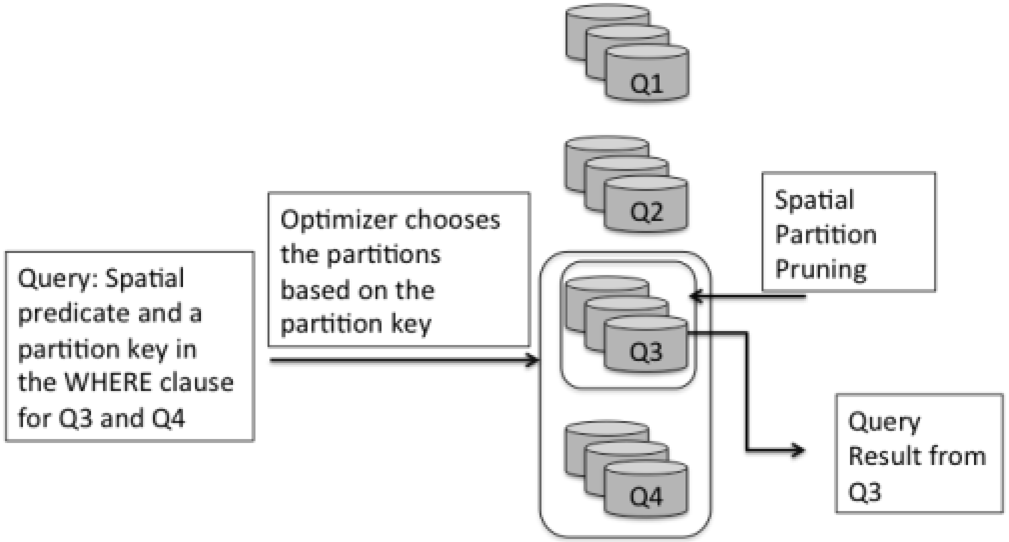
Analysts: Rowan Curran with Holger Kisker, Ph.D. and Emily Miller
September 1, 2016

ORACLE®

12.2 Spatial Feature Overview

- Spatial Index Enhancements
- GeoJSON Support in the DB
- Location Tracking Server
- Data Enrichment API
- Extended 3D Support
- Mapping with the DB Option
- Other New Features

Spatial Index Improvements



Query: Spatial predicate and a partition key in the WHERE clause for Q3 and Q4

Optimizer chooses the partitions based on the partition key

Spatial Partition Pruning

Query Result from Q3

- Support for all partitioning methods
  - Previously restricted to range partitioning
- List partitioning, interval partitioning, hash partitioning, etc.

List Partitioning Example

- Spatial table with data for each country
  - Partition the table so that each country is in its own partition

```
Create table road_segments (…)
Partition by range (country_code)
(Partition p1 values less than ‘B’,
Partition p2 values less than ‘C’,
…. );
```

```
Create table road_segments (…)
Partition by list (country_code)
(Partition p1 values (‘AE’),
Partition p2 values (‘AR’, ‘AS’),
Partition p3 values (‘US’),
…. );
```

Spatial Index Improvements

- Improve Create Index Performance
  - Using GTTs for intermediate steps
  - Reduced redo by a factor of 3
  - 2 to 3 times faster
- Index statistics collection performance
  - Index Stats for 100 million rows
  - Redo: **772k** (183,563,073 MBRs in 512 rows)
  - total collection elapsed time was 00:0**2:20.62** minutes

New Index Type Optimized for Point Data

- R-tree index has performance bottlenecks when large numbers of concurrent DMLs are done on the spatial tables
- Spatial index based on B-tree index
  - Internally create a composite b-tree index on (x,y) or (x,y,z)
- Improves DML performance for large volumes of updates for point only data
- Queries are still done using the same standard syntax, so no application changes are required

```
CREATE INDEX pt_idx on PT_CB(c2) indextype is mdsys.spatial_index_v2
PAREMETERS ('layer_gtype=POINT cbtree_index=true');
```

Spatial Index Improvements

- Composite B-tree based index
- Point data (10 million) Tests
- create index sidx on abi(geometry) indextype is mdsys.spatial_index parameters ('layer_gtype=POINT');
- Index created. **Elapsed: 00:13:15.34**
- SQL> create index sidx on abi(geometry) indextype is mdsys.spatial_index parameters ('layer_gtype=POINT cbtree_index=true');
- Index created. **Elapsed: 00:00:15.40**

Spatial Index Improvements

- Q.1: select count(*) from abi, states where sdo_filter(geometry, geom) = 'TRUE';
  - Composite B-tree / Non-geodetic : Elapsed: 00:00:16.12
  - R-tree / Non-geodetic : Elapsed: 00:00:57.51
  - Composite B-tree / Geodetic : Elapsed: 00:00:15.51
  - R-tree / Geodetic : Elapsed: 00:00:43.30
- Q.2: select count(*) from abi, counties where sdo_filter(geometry, geom) = 'TRUE';
  - Composite B-tree / Non-geodetic : Elapsed: 00:01:06.77
  - R-tree / Non-geodetic : Elapsed: 00:01:09.81
  - Composite B-tree / Geodetic : Elapsed: 00:01:11.83
  - R-tree / Geodetic : Elapsed: 00:00:54.94 2

GeoJSON support

-
  ```
  select json_value('{"type": "Point",
                     "coordinates": [125.6, 10.1]}', '$'
        returning sdo_geometry) from dual
  ```

- Extend JSON support in the database with Spatial operations
  - JSON_VALUE() to support GeoJSON and SDO_GEOMETRY
- SDO_GEOMETRY constructors extended to take JSON as input
- Support spatial index and spatial queries on JSON documents

GeoJSON support

-
  ```
  create index sidx4 on JSON_20893895 (json_value(c2,'$' returning
  mdsys.sdo_geometry))
  indextype is mdsys.spatial_index_v2;
  ```

-
  ```
  select EMPID
     from EMPLOYEES
    where SDO_WITHIN_DISTANCE(
            JSON_VALUE("JSON_VALUE", '$.location[0]' returning
  SDO_GEOMETRY),  SDO_GEOMETRY(2001,8307,
  SDO_POINT_TYPE(74,40,NULL), NULL, NULL))),
           'distance=100 unit=mile') ='TRUE';
  ```
- Default metadata values supported for all JSON tables
  - SRID: 4326, tolerance: 0.05

SDO_GEOMETRY or GeoJSON?

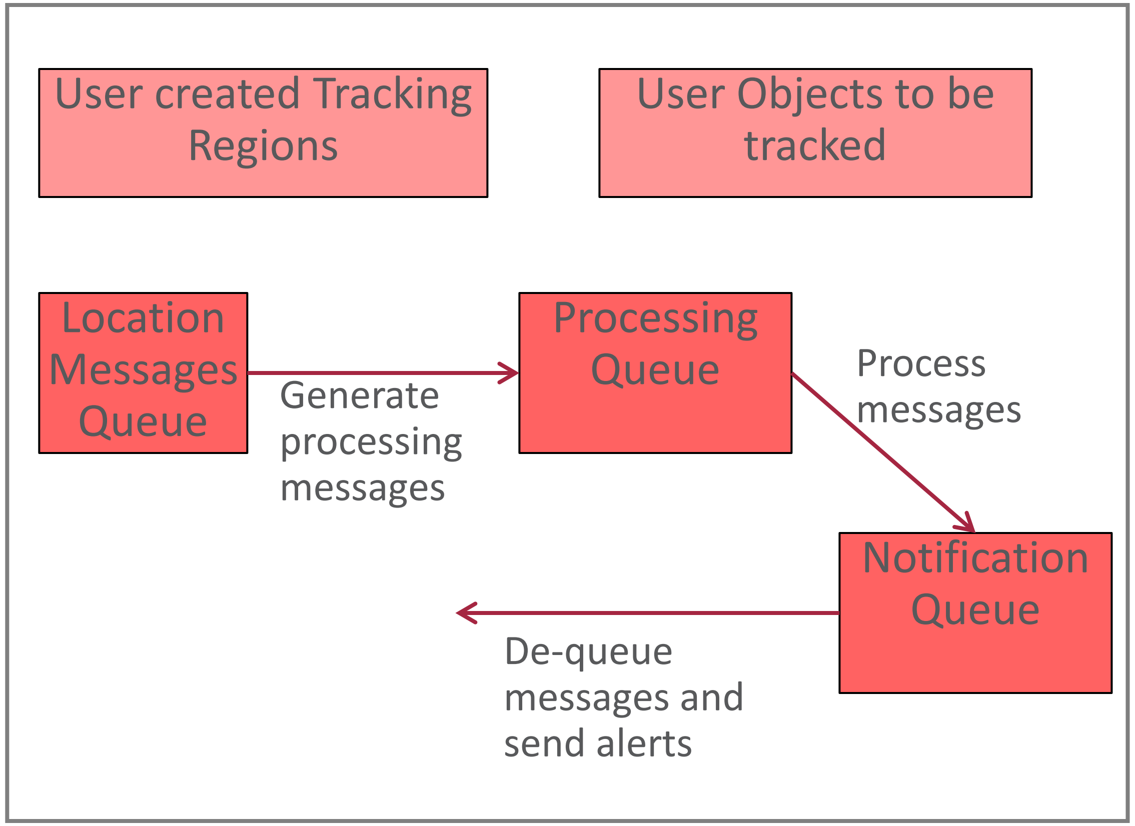
- This is still determined by the application design
  - Application needs a JSON document store, then the geometry attributes are stored as GeoJSON as part of the JSON document store
- Users can mix standard JSON predicates and spatial predicates in the same query
- How to integrate third party applications that expect SDO_GEOMETRY
  - Define a function based index JSON returning SDO_GEOMETRY
  - Convert between JSON and SDO_GEOMETRY using SDO_UTIL functions
    - SDO_UTIL.TO_GeoJSON
    - SDO_UTIL.From_GeoJSON

Standard spatial queries on external tables

- Spatial operations can be performed from the database using SQL
- JOIN between a DB table: CINEMA and external table: TWEETS_EXT_TAB_FILE
- This can be used to integrate queries between BDSG and DB tables

```
select sdo_geom.SDO_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326), 0.05, 'UNIT=MILE'),
         ci.name, tw.user_id
from CINEMA ci, TWEETS_EXT_TAB_FILE tw
where SDO_WITHIN_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326),
                                             'DISTANCE=0.25 UNIT=MILE') = 'TRUE‘
select sdo_geom.SDO_DISTANCE(ci.geometry, SDO_GEOMETRY(tw.geometry, 4326), 0.05, 'UNIT=MILE'),
         ci.name, tw.user_id
from CINEMA ci, TWEETS_EXT_TAB_FILE tw
where SDO_WITHIN_DISTANCE(SDO_GEOMETRY(tw.geometry, 4326),  ci.geometry,
                                             'DISTANCE=0.25 UNIT=MILE') = 'TRUE'
```

Location tracking APIs


User created Tracking Regions
User Objects to be tracked
Location Messages Queue
Generate processing messages
Processing Queue
Process messages
Notification Queue
De-queue messages and send alerts

- Enhanced “point in polygon” analysis
  - Parallel polygon analysis for multiple moving objects and tracking regions
  - track millions of moving objects against thousands of regions
- Designed for cloud based tracking services
- PL/SQL API
- Java API using JMX to capture and track location-based events for very large volumes of moving objects

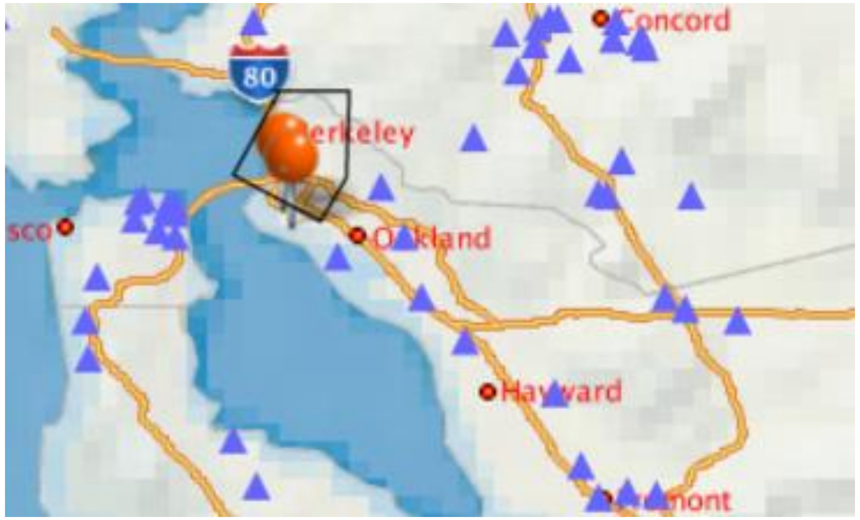
Location Tracking API workflow

- Create regions of interest in the DB as polygon geometries
- Initialize the location tracking server in the DB
  - This will initialize 3 queues: one for receiving location objects, one for receiving location messages, and one for storing the notifications after the locations are processed
  - Each location object (moving object) has many location messages
- PL/SQL APIs to insert location objects and location messages
- Java Applications can use AQ Java API to insert data into the input queues
  - JMX queues used for maximum performance
- Retrieve alerts from the output queue and process as required

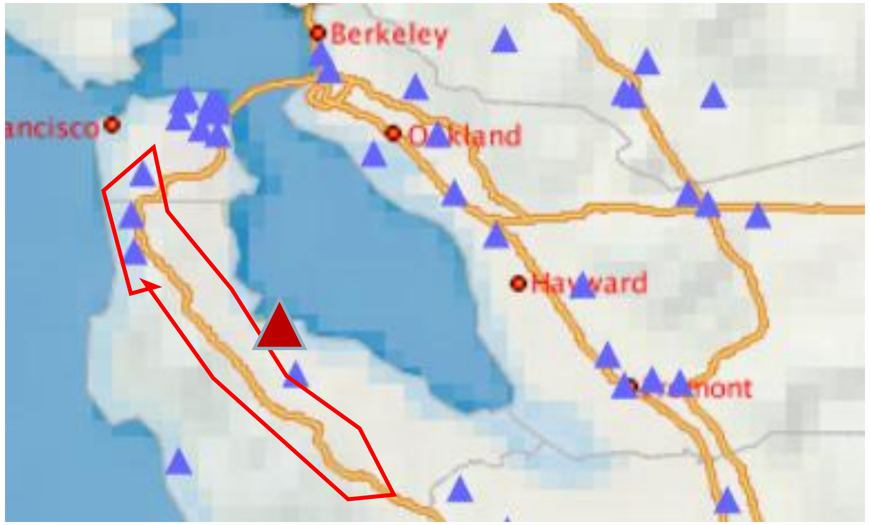
Location tracking Examples

Two types of use cases are supported

- Track objects against a set of polygons and generate notifications when the moving objects enter the regions of interest



- Track objects within a set of polygons and generate notifications when the moving objects leaves the regions of interest

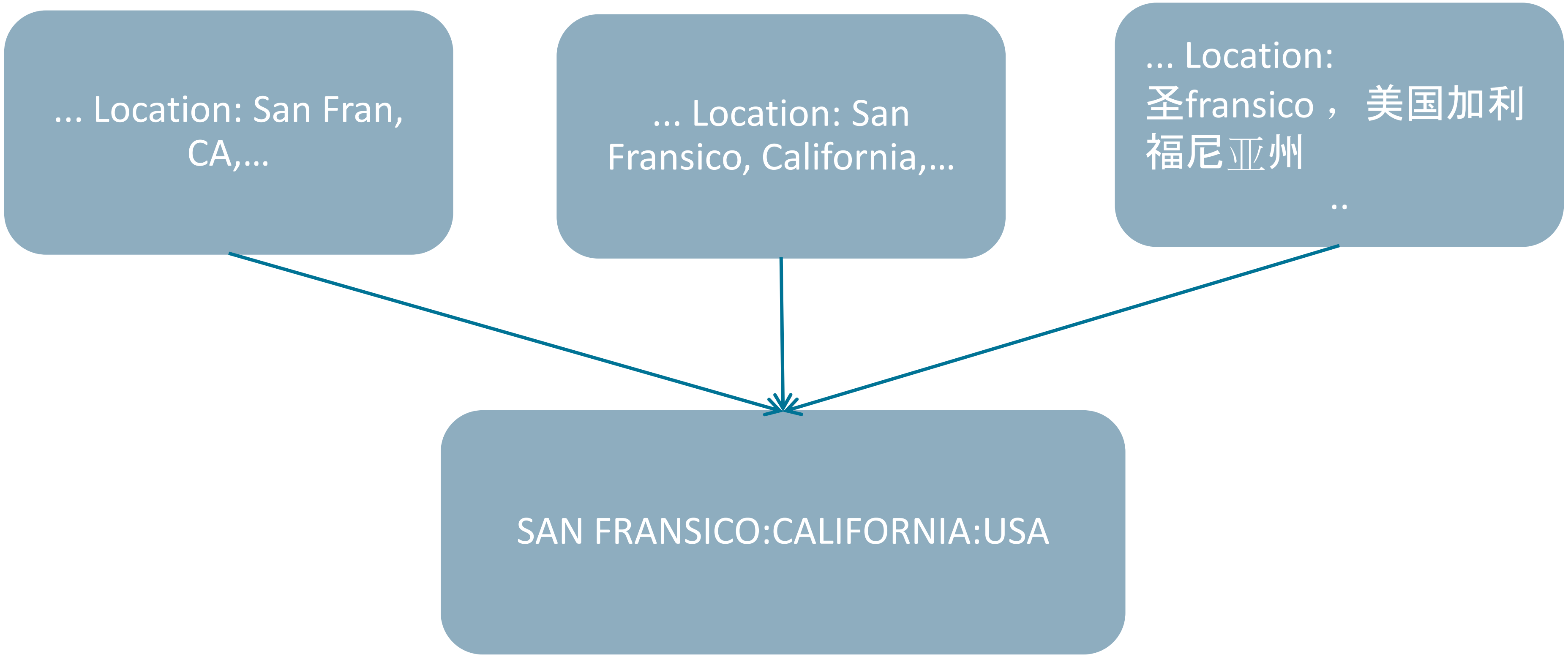


Data Enrichment API

- Search API to match full or partial names with the hierarchy
  - Text Index based POI search for hierarchies
  - For example, “nashua, nh, usa”, “nashua, new hampshire, USA” can match to the same record
- This can be used for data catagerization when large number of records with location tags are collected for analysis
- Hierarchical names data set
  - Contains administrative area names for most of the countries in the world
  - Examples: San Francisco,CA, USA
- This is not part of the geocoder, but stand alone API with bundled data

Use Cases For Data enrichment API

- Call data records or logs from applications
  - Logs can contain references to place names: partial, full or multilingual
  - Match all of these references to the same standard text KEY



... Location: San Fran, CA,...

... Location: San Fransico, California,...

... Location: 圣fransico， 美国加利福尼亚州 ..

SAN FRANSICO:CALIFORNIA:USA

ORACLE®

Examples

```
select * from table(sdo_util.geo_search(‘san fransisco, ca, UNITED STATES'));
```

```
SAN FRANSISCO|SAN FRANCISCO|CALIFORNIA|UNITED STATES ENG -122.4204 37.775592
SOUTH SAN FRANSISCO|SAN FRANCISCO|CALIFORNIA|UNITED STATES ENG -122.4204 37.775592
SAN FRANCISCO|CALIFORNIA|UNITED STATES ENG -122.4204 37.775592
```

```
select * from table(sdo_util.geo_search(‘san frasisco, california, USA'));
```

```
SAN FRANSISCO|SAN FRANCISCO|CALIFORNIA|UNITED STATES ENG -122.4204 37.775592
```

Extending with user defined data

```
insert into ELOC_ADMIN_AREA_SEARCH values (1469286010, 'CENTRAL PARK,NEW YORK CITY,NEW YORK,NYC,RICHMOND,NEW YORK,NY,UNITED STATES,USA', 'CENTRAL PARK', 'CENTRAL PARK|NEW YORK|RICHMOND|NEW YORK|UNITED STATES', 'ENG', 7, 73.9654,40.7829, 0);
```

```
select * from table(sdo_util.geo_search('central park,new york,NY,UNITED STATES'));
```

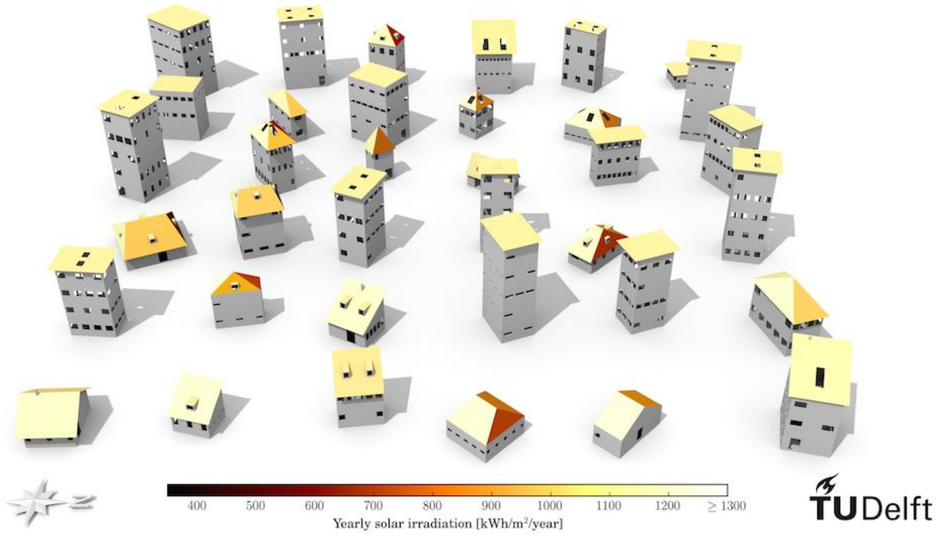
```
CENTRAL PARK CENTRAL PARK|NEW YORK|RICHMOND|NEW YORK|UNITED STATES ENG 73.9654 40.7829
```

Map Visualization with DB Option

- Announcing the bundling of map visualization with DB 12.2 release
- This includes
  - A mapping server (based on our MapViewer)
  - HTML5 API
  - Map Builder
  - Packaged content for admin areas (countries, states, counties as JSON files)
  - OGC Services: WMS, WMTS
- Licensed with Spatial and Graph Option
  - Can be deployed in any WLS
  - On premise or in the Java Cloud Service

City Models – CityGML

- Supports
  - CityGML 2.0
  - 3dCityDB 2.0.1
  - Core model, geometry representation, appearance model, buildings, city furniture model, land use, transportation, vegetation, water body
- Validation
  - Geometric
    - Polygons (SURFACE_GEOMETRY), root geometries
  - Semantic – not yet
- Targeting
  - Large data sets, scalability
  - Data integration
  - Analytical options



GeoRaster 12.2 New Features

1. Native JPEG 2000 Compression
2. New Image Processing Functions
3. New Raster Algebra Functions
4. New Core Functions
5. Performance Improvements
6. Java API and Client-side Viewer and ETL Enhancements

Native JPEG 2000 Compression

- GeoRaster can be natively stored in JPEG 2000 Compression Now
  - The same as JP2 file, stored in a single BLOB and image can be tiled (max 65535 tiles)
  - High compression ratio: For the same quality, 2-7 times smaller than JPEG image
  - Support both lossless and lossy compressions
  - No separate pyramids, pyramid level is limited by tile dimension sizes
  - Support 8 bit and 16 bit integer images
  - No limit on number of bands
  - Support very large size image. Size is limited by memory and max number of tiles (max_mem_size / 20 * 65535)
  - Tiling is recommended
    - Tiling provides scalability and performance

Native JPEG 2000 Compression (Cont.)

- GeoRaster Functionality for JPEG 2000 Compression
  - New procedures:
    - **sdo_geor.compressJP2**
    - **sdo_geor.decompressJP2**
  - Majority of existing GeoRaster functions support reading of JP2 compressed GeoRaster objects, including all key functions such as getRasterSubset, subset, rectify, large scale mosaicking, virtual mosaic, and raster algebra.
  - Most functions, such as getRasterSubset, subset, and virtual mosaic support partial decompressing (faster)
  - A few require full image decompressing (slower)

New Image Processing Functions

- A New Image Processing Package SDO_GEOR_IP
  - linear stretching
  - piecewise stretching
  - equalization
  - normalization
  - histogram matching
  - dodging
  - filtering
  - warping (sdo_geor.warp in SDO_GEOR package)

New Image Processing Functions (Cont.)

- Mosaic Enhancement – New Color Balancing Methods
  - Mosaic Goal: unlimited images, various images, automation, high speed, high quality
  - New color balancing methods drastically improve color quality of mosaics:
    - **Linear Stretching**

      (1) to the given min and max value or (2) to the given reference image band by band
    - **Statistic Matching** (**Normalization**)

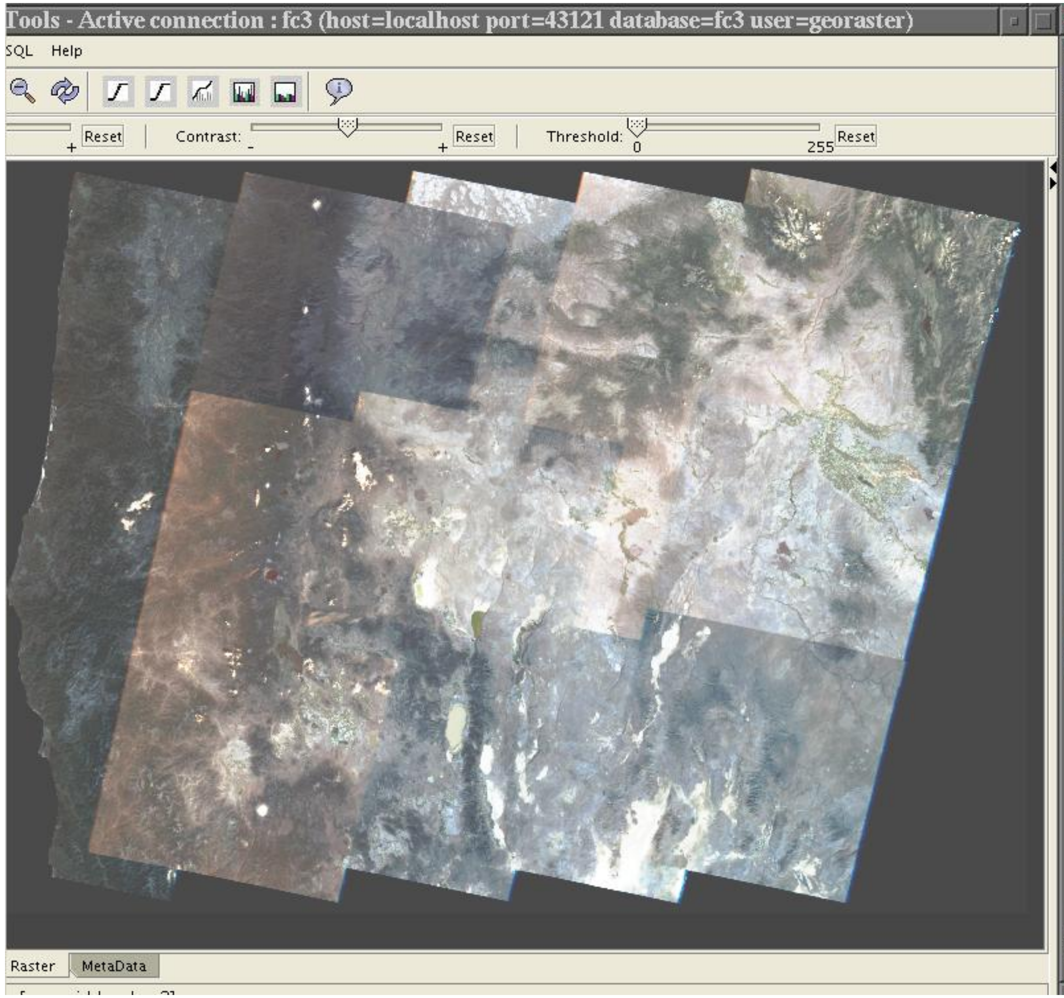
      (1) to the given mean and std value or (2) to the given reference image band by band or

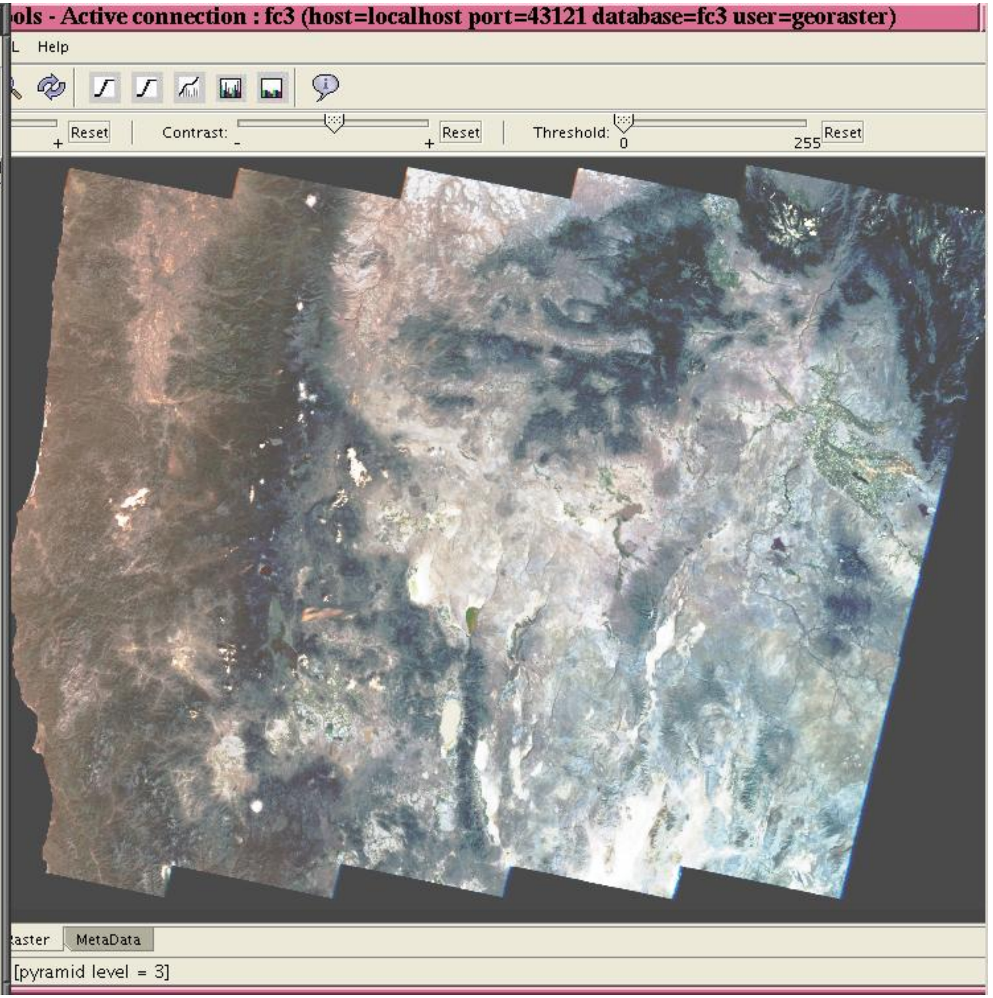
      (3) to use overlapping areas image by image and band by band
    - **Histogram Matching**

      (1) to the given histogram or (2) to the given reference image band by band or

      (3) to use overlapping areas image by image and band by band

Example – Image Mosaic with Histogram Matching


Tools - Active connection : fc3 (host=localhost port=43121 database=fc3 user=georaster)
SQL
Help
Reset
Contrast:
Reset
Threshold:
0
255
Reset
Raster
MetaData


ols - Active connection : fc3 (host=localhost port=43121 database=fc3 user=georaster)
L
Help
Reset
Contrast:
Reset
Threshold:
0
255
Reset
aster
MetaData

[pyramid level = 3]

Virtual Mosaic and Dynamic Tile Layer

Precision Farming Example

- New raster data collected every few weeks
- Application to view the data in a browser
  - Traditional method is to create physical mosaic and then create map tiles
  - Extra storage required and the map tiles need to be updated every time new data is acquired for some farms
- Define virtual mosaics with color balancing
- Display as Dynamic Tile Layers with the Mapping Server
  - Any time new images are acquired for farms, they are automatically used for display

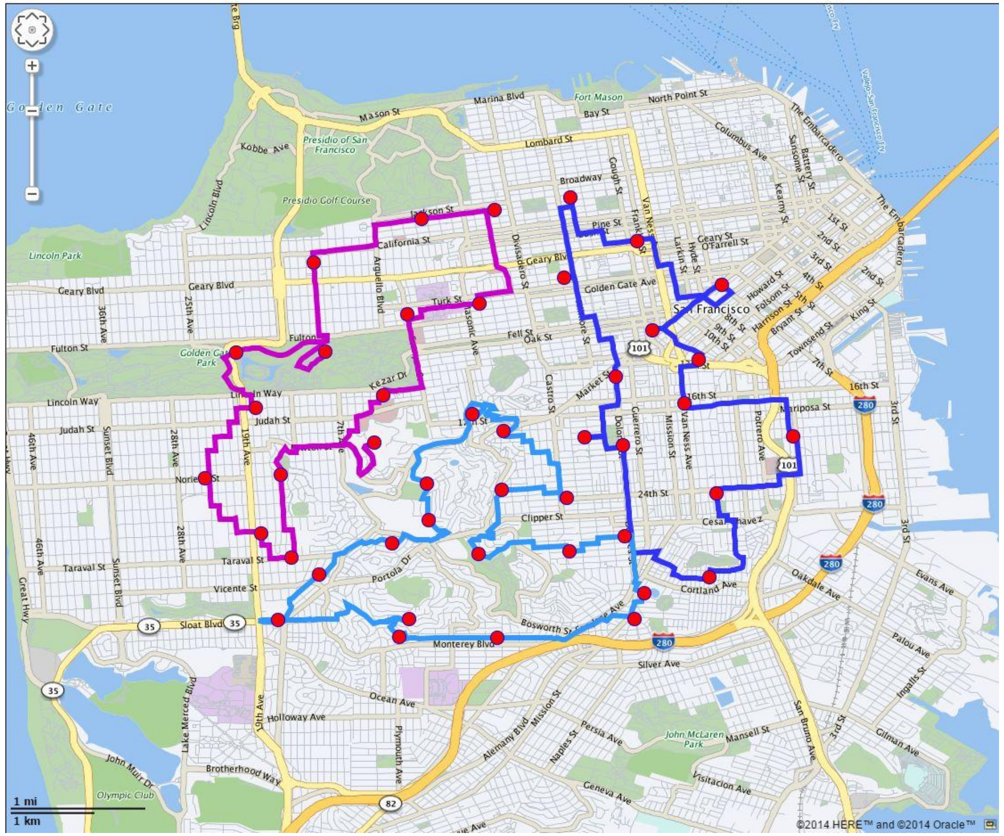
Network Data Model Enhancements

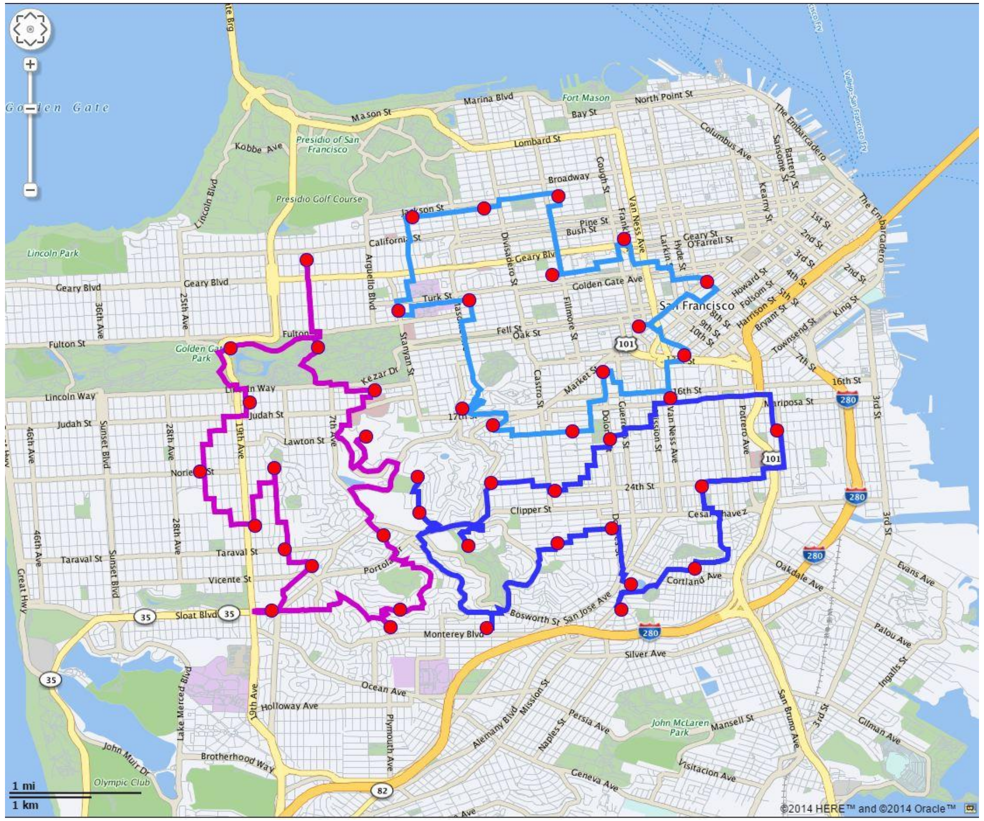
- Multiple TSPs/Single Depot. VRP
- Network Buffers
- Randomized TSP solution
- Feature model editor
- Parallel Framework for Independent Analysis Tasks

Multiple TSPs/Single Depot. VRP

- A solution for multiple sale persons (vehicles)
- Assign each sales person to a group of customers, find the TSP route for each group to minimize total route cost
- Cluster-first, route-second approach
  - Clustering Approaches
    - K Means
    - Bisection Partition
    - User defined Partition approaches
  - TSP route computation
- Can specify a depot for single depot VRP

Clustering/Multiple TSPs Examples





Web Services

- Web catalogue Service 2.0.0.
  - Based on new XML binary storage
  - Supports DCMI, Inspire, ISO profiles
- Web Coverage Service 2.0.0., 2.0.1.
  - Publish data stored in Georaster as coverages
- Improvements for WFS
- Unified console for all web services

Q & A