ORACLE

Oracle Spatial: New Features in Oracle Database 23ai

Maps, Location Intelligence, and Geospatial Platform

What is Oracle Spatial?

Spatial features in converged database

In-database functionality to

- Store and manage all kinds of geospatial data
- Perform spatial analysis where the data resides

Components, APIs & Services

Developer toolbox for

- Map visualization
- Advanced analytics
- Access to spatial functionality and processing workflows

Spatial Studio

Low-code, self-service tool to

- Enable non-experts to more easily analyze data
- Help developers build applications more quickly

What is Oracle Spatial?

Deployment and Licensing

Spatial features in converged database

Integral part of

- Autonomous Database
- ExaCS and ExaC@C
- Database Cloud Service
- Database on-premises

Components, APIs & Services

Deployed on

- OCI Compute
- Available on OCI Cloud Marketplace
- On-premises

Spatial Studio

Deployed on

- OCI Compute
- Available on OCI Cloud Marketplace
- On-premises

New features for Spatial analytics and Geospatial platform

Ease of use

Simpler creation of SDO_GEOMETRY

- Constructor for point data in Longitude/Latitude form
- Constants for geometry type and coordinate system

Simpler creation of spatial index

- Automated creation of index metadata

Spatial Studio

- Integration of published projects into web apps
- Improved interaction with maps, data import from Excel and CSV files

Developer features

REST API for raster data

- Data access
- Processing
- Import/export
- Virtual mosaics

Geocoding PL/SQL API (for ADB-Serverless)

Spatial Studio

- Improved integration of background maps from web services (WMS)

3D data

Point cloud cross section

Point cloud difference

- Change detection

Create mesh from point data

New In 23ai

Ease of use

Simpler syntax to work with longitude/latitude point geometries

You can now use this short, intuitive syntax to construct longitude/latitude points:

```
SDO_GEOMETRY(longitude value, latitude value)     SDO_GEOMETRY(-100.123, 20.456)
```

Replaces longer syntax:

```
SDO_GEOMETRY(2001, 4326, SDO_POINT_TYPE(-100.123, 20.456, null), null, null)
```

Examples

```
INSERT INTO MY_TABLE (GEOMETRY)
VALUES
  ( SDO_GEOMETRY(-100.123, 20.456) );
```

Populate the spatial geometry column in my table with the longitude/latitude point (-100.123, 20.445)

```
SELECT PROPERTY_ID
FROM PROPERTIES p
WHERE SDO_WITHIN_DISTANCE(p.GEOMETRY,
   SDO_GEOMETRY(-100.123, 20.456),
   'distance=30 unit=MILE') = 'TRUE';
```

Find all properties within 30 miles of the point with longitude/ latitude (-100.123, 20.445)

New In 23ai

Ease of use

More intuitive syntax to work with Spatial geometries

Basic Spatial data type structure:

```
SDO_GEOMETRY( SDO_GTYPE , SDO_SRID , SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES )
```

- SDO_GTYPE: What type of geometry is it?
- SDO_SRID: The coordinate system used, such as longitude/latitude
- SDO_POINT, SDO_ELEM_INFO, SDO_ORDINATES: Defines the shape

Now, use → `SDO_GEOMETRY(sdo_polygon2d, sdo_lonlat,...`

Instead of → `SDO_GEOMETRY(2003, 4326,...`

Human-readable constants instead of numeric codes, to describe

- Geometry types: Points, linestrings, polygons, multi-polygons, etc. in 2D or 3D
- 2 common coordinate systems: Longitude / latitude and Web Mercator

New In 23ai

Ease of use

Use in INSERT and WHERE clauses, to define a geometry, or use in spatial query criteria

Examples

```
INSERT INTO MY_TABLE (GEOMETRY) VALUES (
  SDO_GEOMETRY(sdo_polygon2d, sdo_lonlat,
               null, SDO_ELEM_INFO_ARRAY(1,1003,3),
               SDO_ORDINATE_ARRAY(-100.1,20.2,-90.2,23.4)));
```

Create a 2D polygon geometry and insert it into my table

```
SELECT count(*)
  FROM MY_TABLE a
 WHERE a.GEOMETRY.SDO_GTYPE = sdo_polygon2d ;
```

Find how many rows in my table are 2D polygons

New In 23ai

Ease of use

Spatial metadata now automatically created

- To use spatial indexes, Oracle Spatial requires metadata
- Before, you had to manually insert this metadata before creating the index
- Now, spatial metadata is automatically inserted when you create a spatial index
- Very convenient to operationalize the creation and indexing of spatial data

```
INSERT INTO TEST VALUES (1, SDO_GEOMETRY(-73.45, 45.2));
COMMIT;
```

```
INSERT INTO USER_SDO_GEOM_METADATA VALUES ('TEST', 'G',
       SDO_DIM_ARRAY(SDO_DIM_ELEMENT('X', -180, 180, 0.05),
       SDO_DIM_ELEMENT('Y', -90, 90, 0.05)), 4326);
COMMIT;
```

You can now skip this step – since metadata is automatically inserted when you create index

```
CREATE INDEX TEST_SIDX ON TEST(G) INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2;
```

New In 23ai

Enhanced support for raster data

In many industries, customers are using geo-referenced raster data

- Raster images: Data from satellites, surveying aircraft, drones, etc.
- Gridded data: Weather forecasts or other simulations, socio-demographic data, etc.

In Oracle Spatial we have had capabilities to work this kind of data since 2004

- Using SDO_GEORASTER data type to store, index, query, analyze and publish raster data

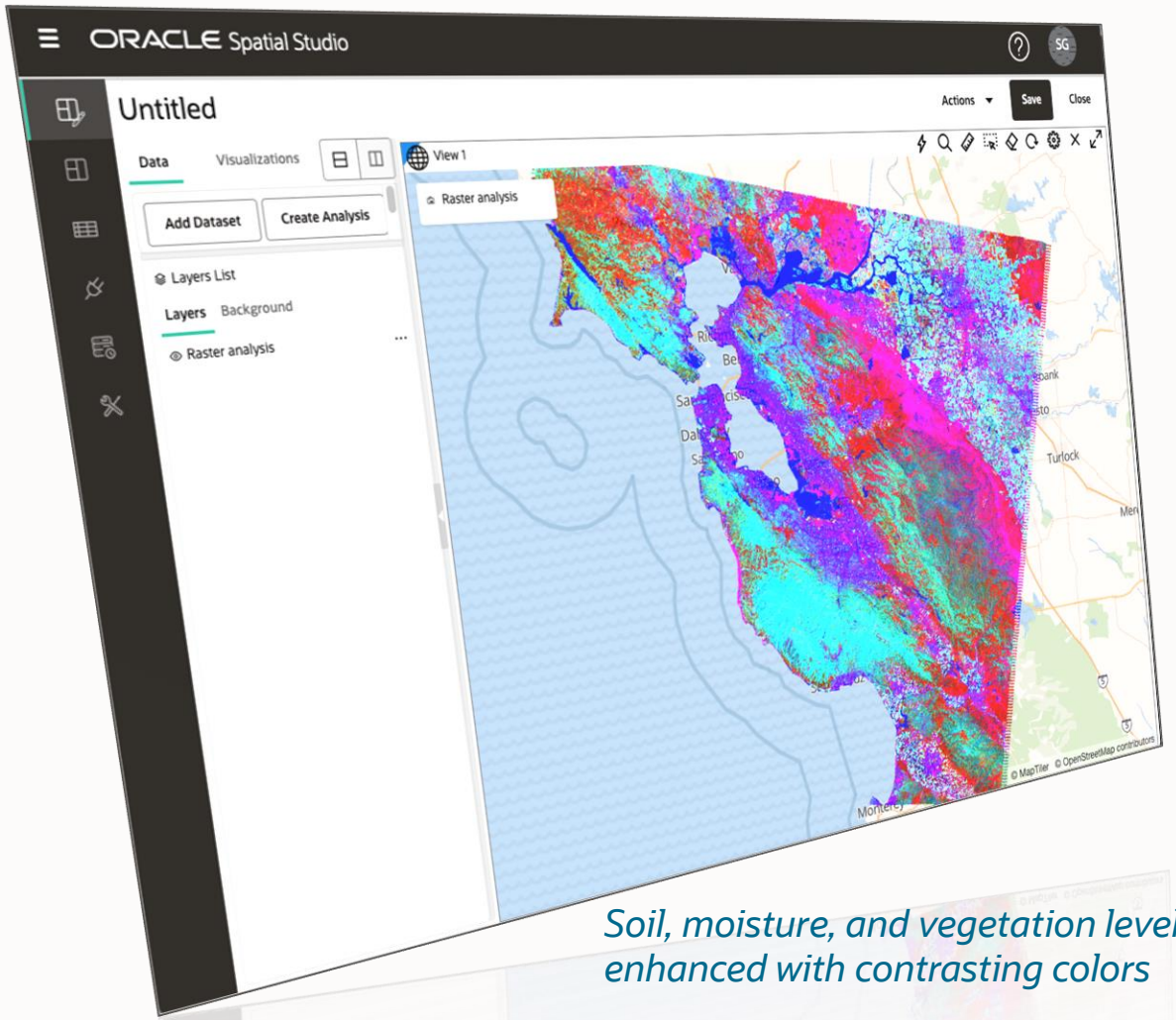
So far, we have offered PL/SQL and Java APIs for these purposes

In Database 23ai, we have enhanced the current functionality by adding a REST API

- Enabling programmatic access to raster functionality from web-based applications
- REST APIs can be deployed in Weblogic or Tomcat
- Also available as OCI Marketplace image

How does Oracle Spatial work?

Support for raster data



Soil, moisture, and vegetation levels, enhanced with contrasting colors

Data type to store georeferenced regular grids of data

- Socio-demographic data
- Weather forecasts

Designed for huge data volumes

Raster data processing, e.g.

- Data extraction by clipping
- Interpolation of cell values

Raster analytics, e.g.

- Statistical operations
- Algebraic expressions

Also usable for other types of raster data, e.g.

- Satellite images, aerial images

New In 23ai

GeoRaster REST API

Overview

Wraps entire GeoRaster PL/SQL API

- Metadata queries and updates
- Data access and modification
- Image processing
- Virtual mosaics

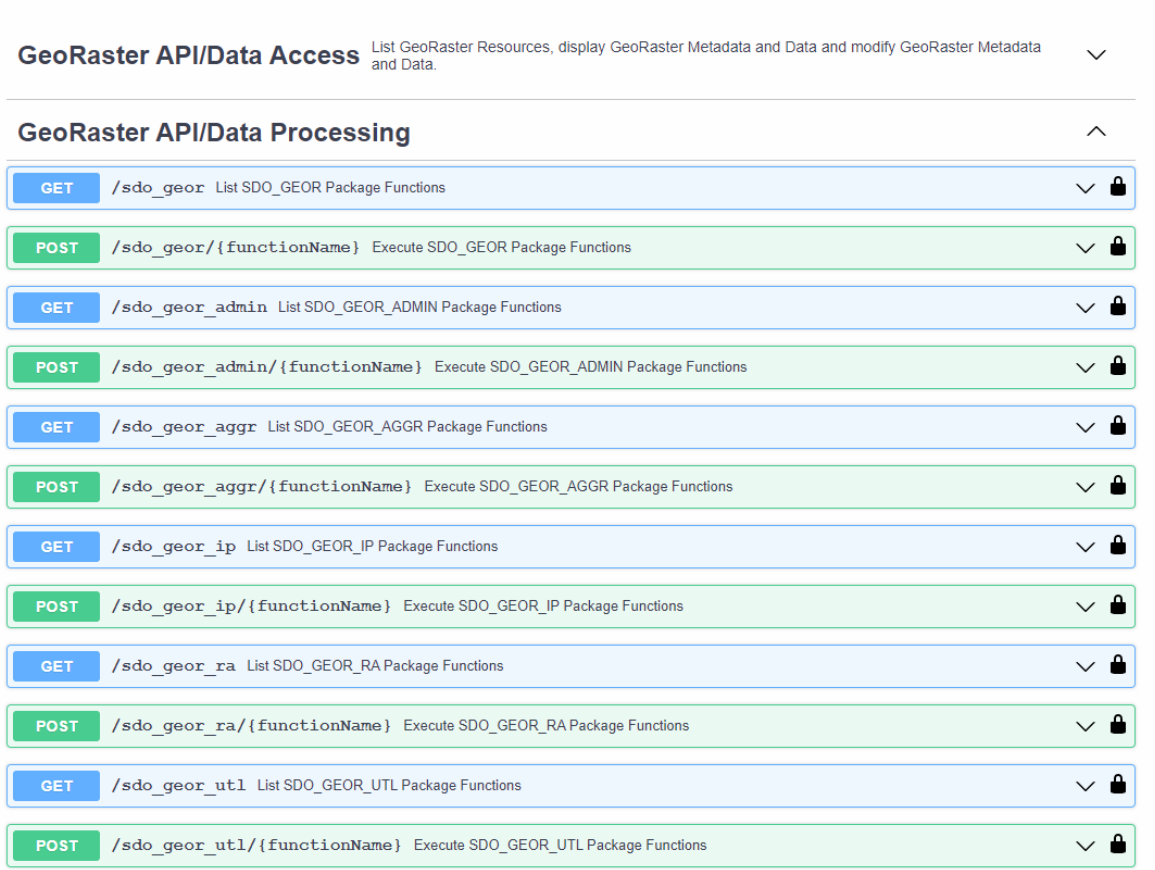
Resulting images can be delivered as image files (PNG, JPEG), or in raw binary format

Support import/export for small raster data files

- Using GDAL to support wide variety of file formats

REST API is bundled with sdows.ear

- Deployment in JEE server in mid-tier



New In 23ai

GeoRaster REST API

Detailed functionality

Data access and processing APIs to execute almost 200 available functions in GeoRaster packages

- Create, modify, and retrieve information about GeoRaster objects
  - Generate resolution pyramid, compress to JPEG2000, generate statistics, etc.
- Image processing functions
  - Filter, normalize, match histograms, stretch, etc.
- Raster algebra functions
  - Classify, perform mathematical operation, stack, find cells, etc.
- Administrative operations
  - List GeoRaster tables, register GeoRaster objects, list dangling GeoRaster data, etc.
- Utility operations
  - Calculate optimal block size, calculate surface area, generate color ramp, etc.

Example – list all resources in dataset1

```
curl https://localhost/oraclespatial/georaster/v1/dataset1
```

New In 23ai

Enhanced support for 3D data

Today, our customers are collecting vast amounts of 3D data using

- Laser scanners (LiDAR)
- Stereo images (photogrammetry)

Both methods generate large volumes of (x,y,z) points, so-called point clouds

- Creating a digital twin of the as-built environment

To date, our focus has been on data management (ingest, store, query, extract)

In Database 23ai, we are enhancing the current functionality for point clouds by adding:

- Change detection between point clouds captured at different times
- Computation of vertical cross sections to perform 2D measurements or analyses on 3D data
- Conversion of point cloud data to meshes to represent 3D structures as objects for visualization

New In 23ai

3D Point Cloud Analysis

Cross Section – `SDO_PC_PKG.GENERATE_CROSS_SECTION_AS_GEOMS()`

- Computing cross section of a point cloud
  - Based on a (vertical) input plane
  - Plane defined by 2D vector
- Result includes points inside a configurable buffer around the plane
  - Allows inclusion of nearby points for more complete outline
- Result tables generated:
  - 3D points in the input plane buffer
  - 2D points on the input plane
  - 2D multipoints on the input plane
- Useful for
  - Determining outlines of objects
  - Measurements inside point clouds

Example of a 3D Point Cloud cross section

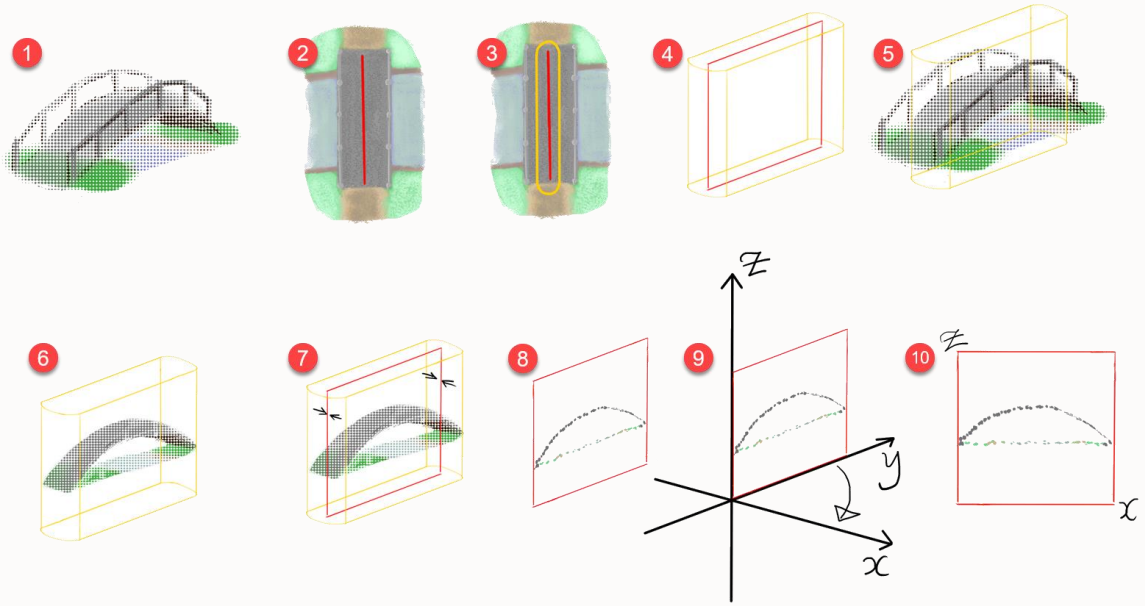


Image courtesy of CenterOne, www.centerone.nl

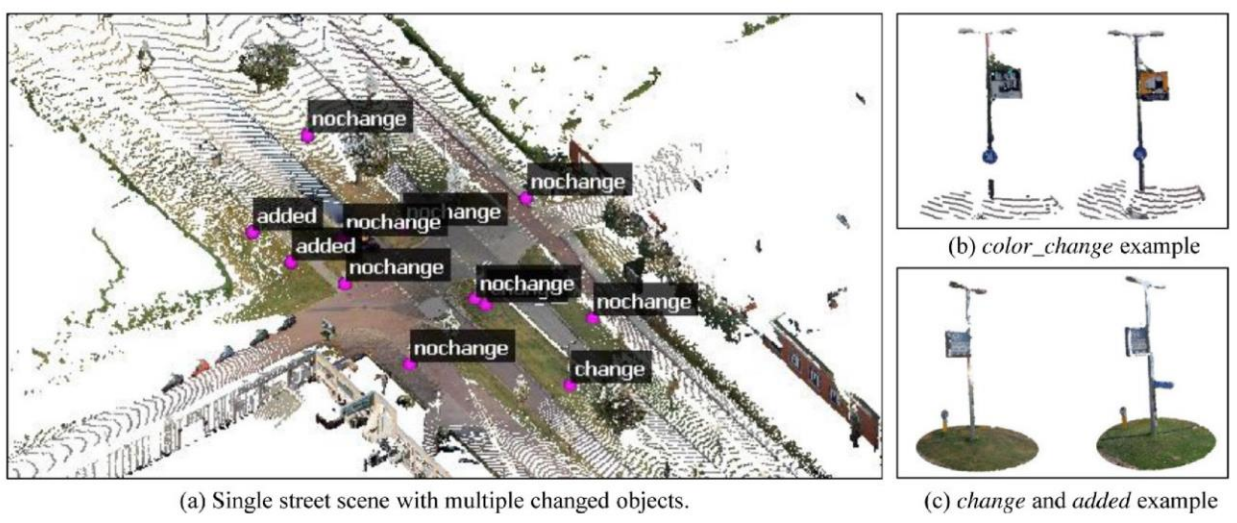
CenterOne

New In 23ai

3D Point Cloud Analysis

Difference – SDO_PC_PKG.PC_DIFFERENCE()

- Computing difference between point clouds captured at different times
  - operates on two point clouds, yielding a third
  - Identifies all points in either without close neighbors in the other
- Useful for change detection



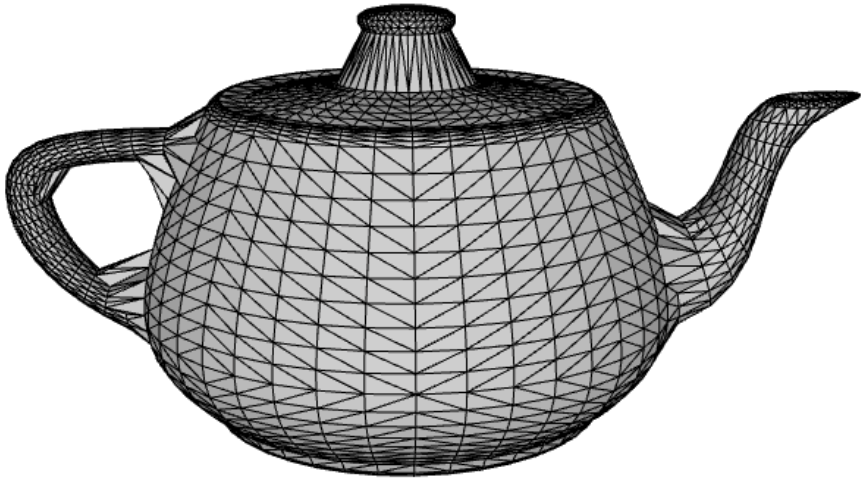
from Three Dimensional Change Detection Using Point Clouds: A Review, Geomatics 2022, doi.org/10.3390/geomatics2040025

3D Mesh generation

Converting point data to meshes – `SDO_TIN_PKG.CREATE_MESHES()`

NEW IN 23c

- TIN (Triangulated Irregular Network) data model extended to support 3D meshes
- Allows generating 3D surfaces from 3D points
  - Similar to TIN generation, but supporting vertical surfaces and overhangs
  - Using table with (x,y,z) as input
  - Result stored in SDO_TIN data type
- Queries on SDO_TIN work the same way for TINs and meshes
- Useful for representation of arbitrary 3D objects



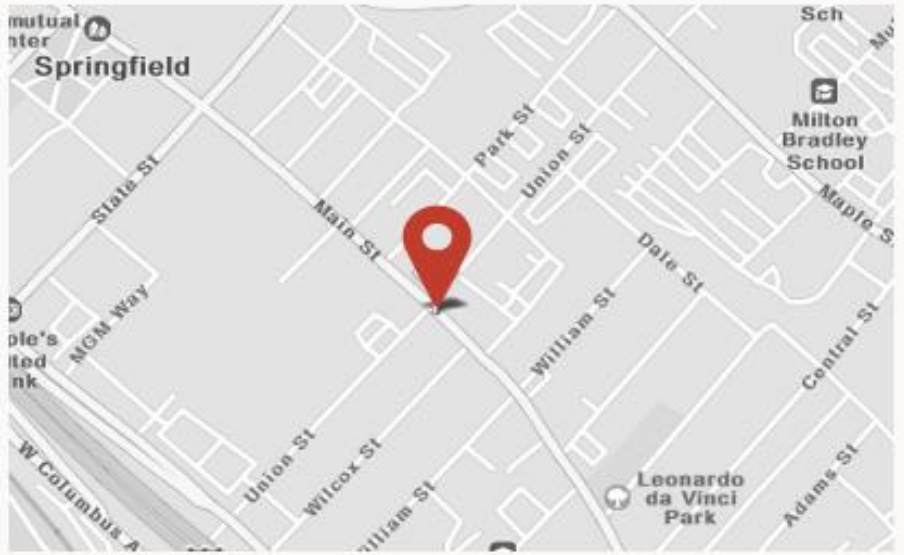
Geocoding PL/SQL API

Available in Autonomous Database-Serverless (ADB-S)

- Geocode your business data to visualize it on maps and perform location analysis
- PL/SQL API to a hosted geocoding service
- Converts structured or unstructured addresses into geographic coordinates
- Location returned in SDO_GEOMETRY or GeoJSON format
- Reverse geocoding (coordinates to addresses) also supported
- Only available in ADB-Serverless
- No need to deploy additional applications or get reference data sets
- **Seamlessly blend geocoding into your database PL/SQL development environment**

993 Main St, Springfield, MA 01103

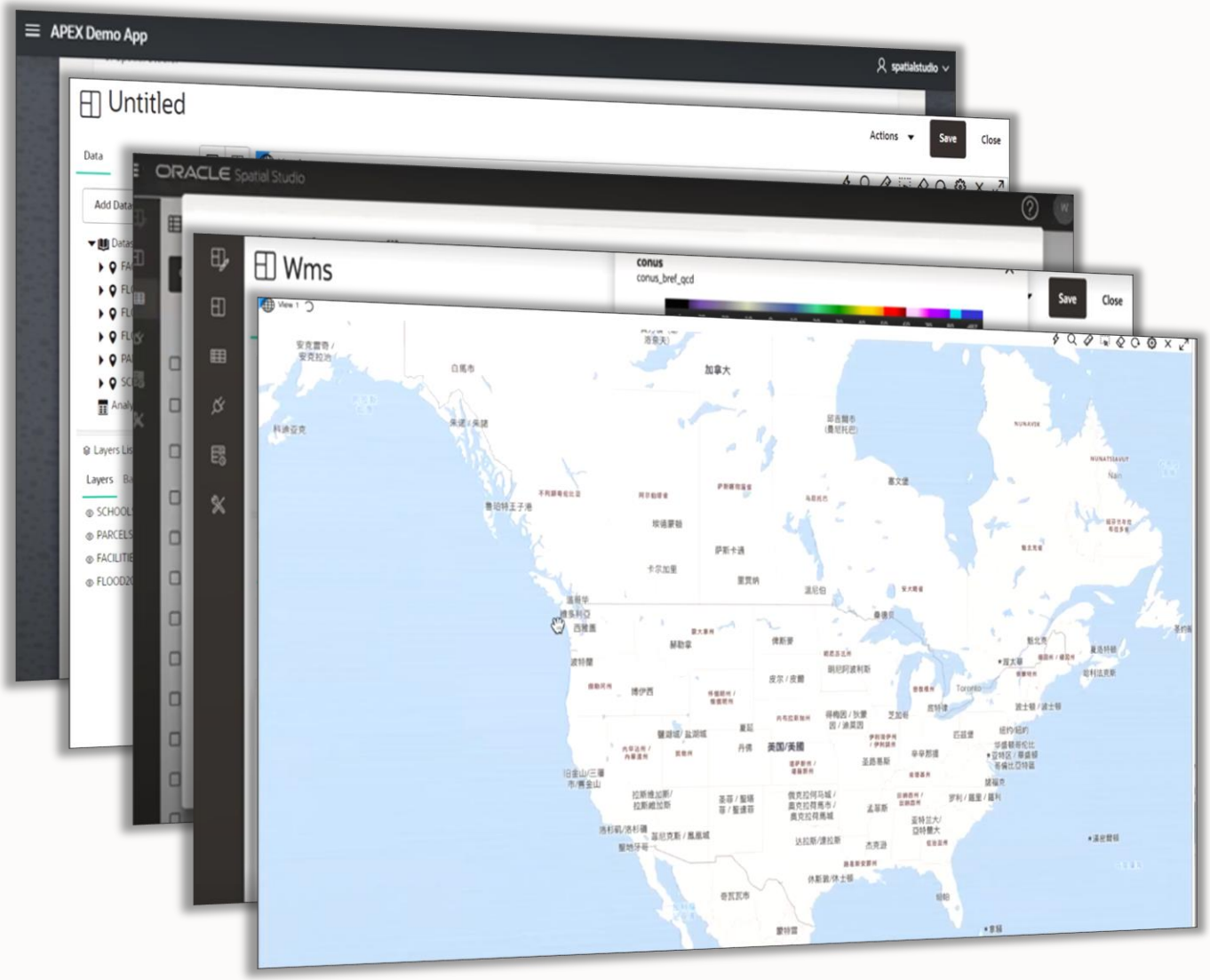
longitude = -72.58435, latitude = 42.09908



Blog: bit.ly/GeocoderADB

LiveLabs Sprint: bit.ly/GeocodingSprint

Spatial Studio 23.1 Enhancements



- **Integration of published projects into web apps**
  - Embed your Spatial Studio project in an external web application or in an APEX Application
- **Improved interaction with maps**
  - The enhanced info window allows you to display data values from overlapping items across map layers
- **Enhanced data import from Excel and CSV files**
  - Importing CSV and Excel files containing WKT (well-known text) and GeoJSON is now supported
- **Better integration with web service (WMS) background maps**
  - View legends when OGC WMS (Open Geospatial Consortium Web Map Service) datasets are visualized on a map
- **Improved multi-lingual support**
  - Vector tile base maps now support displaying map features and labels in different languages

For more information...

- **Oracle Spatial**
  https://www.oracle.com/database/spatial/

- **Blog**
  https://blogs.oracle.com/database/category/db-spatial

- **Oracle Database 23ai Documentation**
  https://docs.oracle.com/en/database/oracle/oracle-database

Oracle LiveLabs

- **Oracle Spatial LiveLabs**
  https://bit.ly/SpatialLiveLabs

- **LiveLabs Sprint – How to geocode addresses in Autonomous Database**
  https://bit.ly/GeocodingSprint

Showcasing how Oracle's solutions solve your business problems