Oracle Business Intelligence 11g Maps:

Making Yourself Spatially Aware

Antony Heljula

30th September 2012
Making Yourself Spatially Aware

**Agenda**

- Overview of Mapping Capability
- Demonstration
- Oracle Spatial Architecture:
  - Oracle MapViewer
  - Oracle MapBuilder
  - Oracle Database
- Steps to Implement
  - Defining Base Maps
  - Defining Layers
  - Map Administration
- Oracle Spatial
- Spatial Analytical Functions
Making Yourself Spatially Aware

Overview of Mapping Capability
Overview of Mapping Capability

Summary

• The ability to embed maps into Oracle BI Dashboards was an exciting new feature introduced with Oracle BI 11g

• Report developers have the power to build colourful and interactive maps within their BI Dashboards....no coding or technical know-how required!
Overview of Mapping Capability

Various Types of “Layer”

- Colour Fill
- Bar Graph
- Pie Graph
- Bubble
- Variable Shape
- Image

© Peak Indicators Limited
Overview of Mapping Capability

**Enabling/Disabling Layers**

- The user can choose which layers to display on the map

Turn off a layer simply by unchecking the corresponding check-box.
Overview of Mapping Capability

*Interaction*

- It is possible to interact with Maps just like you can with other any other OBIEE chart
  - Drill-downs, Action Links, Master-Detail Linking supported

Drill-Down to a lower hierarchy level

Action Link to open up a different analysis
Overview of Mapping Capability

*Interaction Example*

- Multiple layers of map navigation via “Action Links”:
Overview of Mapping Capability

*Rectangle Tool Zoom*

- Clicking on the zoom button 🕵️‍♂️ enables you to zoom to a specific area by drawing a rectangle on the map (OBIEE 11.1.1.6.2. BP1):
Overview of Mapping Capability

*Printing and Downloading*

- Oracle BI 11g supports download and print options for Maps:

  - Powerpoint
  - PDF
Overview of Mapping Capability

**Custom Point Layers**

- "Custom Point Layers" are where the co-ordinates are taken from the Analysis itself
  - Mapping coordinates can come from any type of data source!
  - Longitude/Latitude co-ordinates can be provided in a single column or separate columns
  - Supported with Bubble, Variable Shape and Image layers
Making Yourself Spatially Aware
Demonstration
Making Yourself Spatially Aware

Oracle Spatial Architecture
Oracle Spatial Architecture

*Terminologies*

- When it comes to Business Intelligence, we are interested in overlaying data on top of a map, so we use the terminologies:
  - Background Map
  - Layer

*Base Map* + *Layer* =

- Five map providers are supported
- Layers are typically stored in the Oracle Database
- Points, polygons and lines are supported
Oracle Spatial Architecture

*Base Map Support*

- The following map providers are supported:
  - Oracle Database
  - Web Map Service (WMS)
  - Oracle eLocation
  - Google Maps
  - Bing Maps
Oracle Spatial Architecture

*Types of Layer*

- Oracle BI 11g can support 3 different types of layer:
  - Points: Customer location (long/lat)
  - Polygon: Country boundary
  - Line: A journey

- Your layers typically need to be stored in the underlying database so that they are available for MapViewer to render
Oracle Spatial Architecture

Components

• An Oracle Spatial Architecture will typically consist of the following components:
  • **Oracle Database**
    • Stores the definitions of spatial objects (using special geometry data types)
    • “Locator” database option for querying, filtering, joining spatial data
    • “Spatial” database option for manipulating and aggregating spatial data

• **Oracle MapViewer**
  • A J2EE application used for rendering interactive map images and features-of-interest (FOI). Invoked via a series of XML, Java, PL/SQL and Javascript APIs
  • Configures “Base Maps” to use from one or more different providers
  • Provides a layer of caching

• **Oracle MapBuilder**
  • The development tool for MapViewer
  • A graphical Map builder tool that enables you to define the content and structure of maps and their rendering rules
Oracle Spatial Architecture

Oracle MapViewer

Location, Location, Location
Oracle Spatial Architecture

*MapViewer - Example Base Maps*

- **Oracle Database (NAVTEQ)**
- **Google Maps**
- **WMS (OpenStreetMap)**

Existing map tile layers:

<table>
<thead>
<tr>
<th>Select Name</th>
<th>Data Source</th>
<th>Base map</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORLD MAP GOOGLE</td>
<td>APPSDW</td>
<td>WORLD_MAP_IMG</td>
</tr>
<tr>
<td>OPENSTREETMAP</td>
<td>APPSDW</td>
<td></td>
</tr>
<tr>
<td>WORLD_MAP_OBIEE</td>
<td>NAVTEQ</td>
<td>WORLD_MAP_OBIEE</td>
</tr>
<tr>
<td>WORLD_MAP</td>
<td>NAVTEQ</td>
<td>WORLD_MAP</td>
</tr>
</tbody>
</table>
Oracle Spatial Architecture

Oracle MapBuilder
Oracle Spatial Architecture

*With Oracle BI 11g*

Oracle Business Intelligence 11g

11g Interactive Dashboards

Oracle MapViewer 11g

UK
Germany
France
Spain
......

$10M
$20M
$5M
$7.5M

Base Map Provider

Oracle Database

BI Data Sources

Layers
Making Yourself Spatially Aware

Steps to Implement
Steps to Implement

Overview

• There are 5 main steps needed to build a map report:

  1. Choose a Base Map provider (buy licenses if required)
  2. Configure Base Map in Oracle MapBuilder (internal maps only)
  3. Register Base Map in Oracle MapViewer
  4. Configure geometries for your BI “layers”
  5. Map Administration in Oracle BI
Steps to Implement

Overview

• There are 5 main steps needed to build a map report:

1. Choose a Base Map provider (buy licenses if required)
2. Configure Base Map in Oracle MapBuilder (internal maps only)
3. Register Base Map in Oracle MapViewer
4. Configure geometries for your BI “layers”
5. Map Administration in Oracle BI
Steps to Implement

*Choose a Base Map Provider - Examples*

- **NAVTEQ**
  - NAVTEQ provide “free” World sample map to all Oracle customers
  - Needs to be imported into the Oracle Database
  - More complete/detailed maps have to be purchased

- **GFK GeoMarketing**
  - Provide digital map data based on “Administrative Boundaries” (postal areas)
  - Ideal for when your data includes postal codes / zipcodes
  - Special licensing arrangement for Business Intelligence usage

- **Google Maps**
  - Google Maps are only free if your BI application is on a freely available public web-site! (no username/password required)
  - Google impose license costs for private BI deployments

- **UK Ordnance Survey**
  - Agreement to provide maps to all UK public sector organisations free of charge
  - Free map data available (down to UK postcode level)
  - Detailed UK map data available to private organisations for a license fee
Steps to Implement

*Internal vs External Map Providers*

- Questions are often asked about whether it is better to use “internal” or “external” base maps. But every customer is different!

<table>
<thead>
<tr>
<th>Internal (Oracle DB)</th>
<th>External (e.g. Google Maps)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros:</strong></td>
<td></td>
</tr>
<tr>
<td>• Flexibility to fully configure base map</td>
<td>• Base Maps such as Google are visually pleasing, familiar look and feel</td>
</tr>
<tr>
<td>• No data leaves your network</td>
<td>• Map data is kept up-to-date automatically</td>
</tr>
<tr>
<td>• You are in control of performance</td>
<td>• You are not responsible for Base Map hardware/infrastructure</td>
</tr>
<tr>
<td>• The map data can give you the geometries needed for your BI “layers”</td>
<td>• Easier to standardise on a common map provider within your organisation</td>
</tr>
<tr>
<td>• You can pre-fetch/cache map tiles</td>
<td></td>
</tr>
<tr>
<td><strong>Cons:</strong></td>
<td></td>
</tr>
<tr>
<td>• You are responsible for performance and scalability of Base Map</td>
<td>• Reliant on public internet connection</td>
</tr>
<tr>
<td>• Map data could go out of date if you don’t refresh it</td>
<td>• You have very little control over how the Base Maps are displayed</td>
</tr>
<tr>
<td>• Initial one-off costs could be higher in some cases</td>
<td>• You data leaves your network (you can use secure SSL messaging but Google will still “see” what you are sending)</td>
</tr>
<tr>
<td></td>
<td>• You are not given any geometries which can be use for your BI “layers”</td>
</tr>
<tr>
<td></td>
<td>• No caching of map tiles</td>
</tr>
</tbody>
</table>
Steps to Implement

**Overview**

- There are 5 main steps needed to build a map report:

  1. Choose a Base Map provider (buy licenses if required)
  2. Configure Base Map in Oracle MapBuilder (internal maps only)
  3. Register Base Map in Oracle MapViewer
  4. Configure geometries for your BI “layers”
  5. Map Administration in Oracle BI
Steps to Implement

Configure Base Map in Oracle MapBuilder

- If you have purchased digital map data then you will have import it into your Oracle Database
- If you wish to do so, you can use Oracle MapBuilder to modify the styles on the maps or you could even configure your own base maps

The default world map provided by NAVTEQ comes with 100s of layers. But in this example we have chosen to build our own base map with just two layers: Oceans and Countries. It is still using NAVTEQ digital map data, but it will be faster performing in OBIEE if we are only reporting down to “country” level.
Steps to Implement

*Overview*

- There are 5 main steps needed to build a map report:
  
  1. Choose a Base Map provider (buy licenses if required)
  2. Configure Base Map in Oracle MapBuilder (internal maps only)
  3. Register Base Map in Oracle MapViewer
  4. Configure geometries for your BI “layers”
  5. Map Administration in Oracle BI
Steps to Implement

Register Base Map in Oracle MapViewer

• Once you have your internal/external Base Map available, you need to create a “Map Tile Layer” in Oracle MapViewer
• A Map Tile Layer is effectively a run-time instance of a Base Map, you can specify zoom levels, background colours, image formats etc
• Within MapViewer you can also view and cache your maps, plus you can perform administrative functions such as purge caches
Steps to Implement

Register Base Map in Oracle MapViewer

- When you “Create” a new Map Tile Layer you can choose your type of Base Map:
  - Internal  
    stored inside Oracle Database
  - External  
    e.g. Web Map Service (WMS)
  - Oracle eLocation  
    Oracle’s own map service
  - Bing Maps
  - Google Maps
Steps to Implement

*Overview*

- There are 5 main steps needed to build a map report:

  1. Choose a Base Map provider (buy licenses if required)
  2. Configure Base Map in Oracle MapBuilder (internal maps only)
  3. Register Base Map in Oracle MapViewer
  4. Configure geometries for your BI “layers”
  5. Map Administration in Oracle BI
Steps to Implement

*Configure Geometries for your BI “Layers”*

- Regardless of your Base Map provider, you will typically need to have geometries representing the BI “layers” that you wish to overlay on top of the Base Map.

- Each layer requires two pieces of information:
  - **Key column** To uniquely identify the geometry (and link to Oracle BI)
  - **Geometry** Stored in a column with data type `SDO_GEOMETRY`

- In the example below, we have a table containing a record for every “Account Name” together with its Long/Lat Geometry:

```sql
SQL> SELECT ACCOUNT_NAME, GEOMETRY FROM WH_MAP_CUSTOMER;

<table>
<thead>
<tr>
<th>ACCOUNT_NAME</th>
<th>GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_Northwest Systems</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.67492, 47.37791, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>SSC</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(19.26741, 47.03181, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>Advantage Corp HU</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(19.31172, 47.89269, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Asia</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(19.20336, 47.39532, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Hardware Manufacturing</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.574, 47.3551, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Telecommunications</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.73317, 47.44379, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Application Software</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.7943, 47.6718, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>Tuner and Roberts Inc.</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.62815, 47.08409, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>Stirus Tech</td>
<td>SDO_GEOMETRY(2001, 8307, SDO_POINT_TYPE(18.89278, 47.85304, NULL), NULL, NULL)</td>
</tr>
</tbody>
</table>
```
Steps to Implement

SDO_GEOMETRY Data Type

- SDO_GEOMETRY is the “spatial” data type provided by Oracle, you can store just about any type of geometry within it.

**NOTES**
- When you create an SDO_GEOMETRY column you must create a “spatial” index on it and register the column in the table USER_SDO_GEOM_METADATA.
Steps to Implement

*Where Do You Get The Geometries – Points?*

- In our example we already had a table containing the Long/Lat of every customer account
- So we simply used a SQL update statement to populate the GEOMETRY column:

<table>
<thead>
<tr>
<th>ACCOUNT_NAME</th>
<th>LONGITUDE</th>
<th>LATITUDE</th>
<th>GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_Northwest Systems</td>
<td>18.6792</td>
<td>47.37791</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(18.6792, 47.37791, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>SSC</td>
<td>19.2654</td>
<td>47.05318</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(19.2654, 47.05318, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>Advantage Corp HW</td>
<td>19.3117</td>
<td>47.89269</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(19.3117, 47.89269, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Asia</td>
<td>19.2033</td>
<td>47.39532</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(19.2033, 47.39532, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Hardware Manufacturing</td>
<td>18.7317</td>
<td>47.44379</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(18.7317, 47.44379, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Telecommunications</td>
<td>18.7743</td>
<td>47.6716</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(18.7743, 47.6716, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>XYZ Application Software</td>
<td>18.5281</td>
<td>47.68409</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(18.5281, 47.68409, NULL), NULL, NULL)</td>
</tr>
<tr>
<td>Stirus Tech</td>
<td>18.0927</td>
<td>47.85304</td>
<td>SDO GEOMETRY(2001, 8307, SDO POINT TYPE(18.0927, 47.85304, NULL), NULL, NULL)</td>
</tr>
</tbody>
</table>

- If, however, you only have a Post Code (Zip Code) for each customer account, then you will need to obtain the Long/Lat coordinates of every Post Code – you may have to purchase this from a map provider
Steps to Implement

*Where Do You Get The Geometries – Polygons?*

- If your polygon layers such as “Country Boundaries” need to be stored in the Oracle Database, how do you obtain the geometry data for each country?

- **Answer:** You can simply copy them from your Base Map!

- **Country Boundaries**
- **Ocean Boundaries**
- City Locations
- Highways
- State Boundaries
- Lakes
- etc
Steps to Implement

*Where Do You Get The Geometries – Polygons?*

- In this customer example, we had the “Post Code” for each customer e.g. “SN3 5GH” but we had no geometry information
- GFK GeoMarketing provided world maps containing the boundaries of every postal area/sector
- So we could take the first component of each Post Code e.g. “SN3” and then render sales data on a map in Oracle BI
Steps to Implement

Create “Theme” Within MapBuilder

- Once you have obtained the data for your BI “layer”, you should create a corresponding “Theme” within MapBuilder.
- You can configure styling, labelling and also specify the “Key Column” which will be used to integrate with Oracle BI:

In this example, our Theme actually shows all the Long/Lat locations of our customer accounts.
Steps to Implement

Overview

• There are 5 main steps needed to build a map report:

1. Choose a Base Map provider (buy licenses if required)
2. Configure Base Map in Oracle MapBuilder (internal maps only)
3. Register Base Map in Oracle MapViewer
4. Configure geometries for your BI “layers”
5. Map Administration in Oracle BI
Steps to Implement

*Map Administration in Oracle BI*

- You can access the “Manage Map Data” link via the Administration screen in Oracle BI 11g:
Steps to Implement  
*Manage Map Data - Layers*

- First of all, you define one or more “Layers” that can be rendered on top of a map. Layers can be either:
  - Points
  - Polygons
  - Lines
Steps to Implement

*Manage Map Data - Layers*

- Once the Layer is imported, you can edit/configure it
  - Specify Geometry Type (Either Point, Polygon, or Line)
  - Specify which Subject Area columns can be used for this layer (must join to “Key” column on underlying theme)

In this example, the underlying Theme in MapBuilder is keyed on “ACCOUNT_NAME”

And this key column matches to the “Customer Name” column in our Subject Areas.

So we can now plot “Customer Names” on a map!
Steps to Implement

Manage Map Data - Background Maps

- Once your Layers have been defined, you can import Background Maps and then configure which Layers can use them:
Steps to Implement

Manage Map Data - Background Maps

- Once the Background Maps is imported, you can edit configure it:
  - Specify which Layers can use the map, and at which zoom levels
  - Configure the order in which the Layers appear on top of each other on a map

This Background Map has 3 Layers, with OBIEE_CUSTOMERS displayed on top of the others.
Steps to Implement

*Manage Map Data - Images*

- You can import your own custom Images for use on your maps
- You can import the same images which are defined in Oracle MapBuilder (i.e. the images which are used on your digital base maps)
Steps to Implement

Overview

• That’s it!! You are now ready to build your map report....

✔ Choose a Base Map provider (buy licenses if required)
✔ Configure Base Map in Oracle MapBuilder (internal maps only)
✔ Register Base Map in Oracle MapViewer
✔ Configure geometries for your BI “layers”
✔ Map Administration in Oracle BI
Making Yourself Spatially Aware

Oracle Spatial
Oracle Spatial

Customer Problem

- We already have “country” geometries for each of our Organizations (taken from our base map provider)
- But how do we render the higher-level “Business Groups” on a map?
Oracle Spatial

The Answer

- We use a “spatial aggregate” function. You can aggregate geometries just like you can aggregate sales data:

```sql
SELECT BUSINESS_GROUP, 
       SDO_AGGR_UNION(SDOAGGRTYPE(O.GEOMETRY, 0.1)) GEOMETRY
FROM   WH_BUSINESS_GROUP B, WH_ORGANIZATION O
WHERE  B.BUSINESS_GROUP = O.BUSINESS_GROUP
GROUP BY B.BUSINESS_GROUP;
```
Oracle Spatial

*Oracle Locator vs Oracle Spatial*

- **Oracle Locator**
  - Locator is a cut-down version of Oracle Spatial and is provided free with the database Standard and Enterprise Editions
  - It provides as set of spatial capabilities for application developers:
    - Spatial indexing capabilities
    - Functions for joining and filtering spatial data
    - General utilities e.g. validating spatial data
  - Locator should be sufficient for most of our BI implementations unless, for example, the digital map data being used at the customer site needs to manipulated or aggregated in some way

- **Oracle Spatial**
  - Spatial is a licensed database option and is only available with Enterprise Editions
  - It provides the Locator functionality plus all geometry functions for spatial aggregation, 3D spatial, spatial analysis and mining functions, linear referencing system support and geo-coding
Oracle Spatial

*Oracle Locator vs Oracle Spatial*

- Locator and Spatial functions/objects are created within the “MDSYS” schema on the Oracle database

- It is not immediately obvious which features are Locator (free) and which require a Spatial license

- It is important therefore to refer to the following document which outlines the Locator/Spatial features:
  - [http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_locator.htm#i632018](http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_locator.htm#i632018)

- As a general rule: If you are manipulating spatial data in any way then you probably require a Spatial license!


# Oracle Spatial

**Functionality requiring license Oracle Spatial**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Described in</th>
</tr>
</thead>
<tbody>
<tr>
<td>The following SDO_GEOM package subprograms: SDO_GEOM.RELATE,</td>
<td>Chapter 24</td>
</tr>
<tr>
<td>SDO_GEOM.SDO_DIFFERENCE, SDO_GEOM.SDO_INTERSECTION, SDO_GEOM.SDO_UNION,</td>
<td></td>
</tr>
<tr>
<td>SDO_GEOM.SDO_VOLUME, SDO_GEOM.SDO_XOR,</td>
<td></td>
</tr>
<tr>
<td><strong>SDO_AGGREGATE</strong> spatial aggregate function</td>
<td>Chapter 20</td>
</tr>
<tr>
<td>Linear referencing system (LRS) support</td>
<td>Chapter 7 (concepts and usage) and Chapter 25 (reference)</td>
</tr>
<tr>
<td>Three-dimensional geometry support: the use of 3D spatial indexing, 3D</td>
<td></td>
</tr>
<tr>
<td>operators, and subprograms on 3D data is not supported for Locator.</td>
<td>Section 1.11 (3D concepts and usage)</td>
</tr>
<tr>
<td>Spatial analysis and mining subprograms (SDO_SAM package)</td>
<td>Chapter 29</td>
</tr>
<tr>
<td>OpenLS support, including support for geocoding, mapping, business</td>
<td>Chapter 14, &quot;OpenLS Support&quot; and Chapter 27, &quot;SDO OLS Package (OpenLS)&quot;. See also:</td>
</tr>
<tr>
<td>directory (Yellow Pages), and driving directions (routing) services</td>
<td></td>
</tr>
<tr>
<td>Web feature service (WFS) support (SDO_WFS_PROCESS and SDO_WFS_LOCK packages)</td>
<td>Chapter 15 (concepts and usage), and Chapter 34 and Chapter 33 (reference)</td>
</tr>
<tr>
<td>Catalog services for the Web (CSW) support (SDO_CSW_PROCESS package)</td>
<td>Chapter 16 (concepts and usage) and Chapter 22 (reference)</td>
</tr>
<tr>
<td>Triangulated irregular network (TIN) and point cloud (PC) data types</td>
<td>Section 1.11 (concepts and usage), and Chapter 30 (SDO_TIN_FKG reference) and Chapter 28 (SDO_PC_FKG reference)</td>
</tr>
<tr>
<td>and related subprograms</td>
<td></td>
</tr>
<tr>
<td>GeoRaster support</td>
<td>Oracle Spatial GeoRaster Developer’s Guide</td>
</tr>
<tr>
<td>Topology data model</td>
<td>Oracle Spatial Topology and Network Data Models Developer’s Guide</td>
</tr>
<tr>
<td>Network data model</td>
<td>Oracle Spatial Topology and Network Data Models Developer’s Guide</td>
</tr>
<tr>
<td>Classes in packages other than the oracle.spatial.geometry(adoapi.jar) and oracle.spatial.util(sdoutil.jar) packages of the Spatial Java API</td>
<td>Oracle Spatial Java API Reference</td>
</tr>
</tbody>
</table>
Making Yourself Spatially Aware

Spatial Analytical Functions
Spatial Analytical Functions

Overview

- Oracle BI does not ship with its own “spatial analytical” functions but it is still possible to incorporate them into your BI analyses.
- The example below shows both calculations and filters which involve spatial calculations (e.g. distance between two points).
Spatial Analytical Functions

*Oracle BI “Evaluate” Function*

- In this case, we have used an EVALUATE function to call a database function called “OBIEE_DISTANCE” and pass in two parameters:
  - A location (populated by a OBI Presentation Variable e.g. LONDON)
  - Customer Name

- The PL/SQL function on the database will then return the distance between the two locations (it will run for every record on the report)

<table>
<thead>
<tr>
<th>Customer Name</th>
<th>Total Revenue (Millions)</th>
<th># Orders</th>
<th>Distance From Your Location (KM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul Hunter 1</td>
<td>1.59</td>
<td>23</td>
<td>6.24</td>
</tr>
<tr>
<td>MyTel 3</td>
<td>1.25</td>
<td>23</td>
<td>3.70</td>
</tr>
<tr>
<td>Department of Justice</td>
<td>1.08</td>
<td>22</td>
<td>5.77</td>
</tr>
<tr>
<td>Survey</td>
<td>1.07</td>
<td>17</td>
<td>4.04</td>
</tr>
<tr>
<td>Mike Johnson</td>
<td>1.07</td>
<td>27</td>
<td>2.63</td>
</tr>
<tr>
<td>Carla Johnson 3</td>
<td>0.97</td>
<td>20</td>
<td>8.17</td>
</tr>
</tbody>
</table>

**Column Formula**

```
EVALUATE('obiee_distance("@{PR_LOCATION}@{LONDON}",%1) AS DOUBLE, "Customer"."Customer Name")
```
Questions?
Helping Your Business Intelligence Journey