Spatial

The Final Frontier?
Topics

- Introduction to Spatial concepts
- What's the point (no pun intended)?
- Before you start...
- Some Useful Spatial Data Sources
- Spatial function types
- Usage examples
- Real world customer use cases
Geometry Types

- Polygons
  - useful for defining a geographical boundary or region - from the outline shape of a whole country, to individual states, regions, or custom areas within it.

- Points
  - useful for defining the precise location of a customer's address, a building, or a point of interest

- LineString
  - useful for defining roads, rivers, or the location of underground cables/pipes
Co-ordinate systems and Datums

- Geocentric (used by GPS), based on an (X,Y,Z) co-ordinate system with the origin at the centre of the earth
- Spherical (most well known), based on angles relative to a prime meridian and Equator (Latitude and Longitude)
- Cartesian, defined as a ‘flat’ co-ordinate system placed on a relatively small subset of the surface of Earth
- Many Datums, due to earth not being perfectly round, different approximations of the centre of the earth, continental drift, and historical shortcomings (e.g. WGS84 used by GPS doesn’t include any points in Europe)
Geospatial Grid Index

- A geospatial grid is a simple grid overlay on a geographic area.
- A unique numeric value is assigned to each grid cell to enable queries to quickly identify locations that fall within, or near, a specific geospatial location—for example, site A is within 4 miles of site B.
What's the Point (no pun intended)?

- to combine external with internal information to derive greater insights and business benefits that would not be possible to achieve using internal data alone
  - e.g. store location planning, cannibalisation, market share analysis
- use location-based information to better serve existing customers and generate new revenue streams
  - e.g. new categories of products and services that are location targeted
Before you start...

- Consider upgrading to NPS 7.1 and INZA 3.0 (if not already). Here’s why:
  - Dynamic scheduler rules for WLM; Short query prioritization; Snippet Result Cache; Faster Bulk Fetching with ODBC; Password aging and expiry; nzPortal enhancements; Cryptographic Standards (s800-131a); Support for Replication v1.5
  - Multiple Schema (3-part naming) support; NOT IN / EXIST improvements; CASE WHEN improvements; Truncate table in TXN
  - Faster rebalance for failed disks; Disk validation support; Large scale disk replacement; Call Home v1.0; Enhanced System Health Checks v2.2; IBM License Metric Tool support for “Growth on Demand” pricing

- Consider how you are going to:
  - acquire, transform, load, and visualise spatial data
  - what type of insights and business value a spatial capability could provide
Some Useful Spatial Data Sources

  - provides (anonymised) individual and household level variables by geographical region including: age, gender, occupation, industry, total and median household income, etc.

  - ready made spatial-centric data models by industry/function

- [http://www.gadm.org](http://www.gadm.org)
  - downloadable geodatabases that provide outline polygons for country regions

- [https://koordinates.com](https://koordinates.com)
  - fantastic resource with lots of useful downloadable data sets
Spatial function usage examples

- **Spatial Containment Queries**
  - determine whether a point location (latitude, longitude) falls inside a zone, boundary, or polygon as defined by a series of latitude-longitude pairs, or vertices. For example:
    - Is the dropped call in a coverage area or not?
    - Is an insured property within an earthquake/tsunami/hurricane zone?
    - What marketing zone(s) are my customers located in?
    - ```sql
      SELECT POINTS.LATITUDE, POINTS.LONGITUDE, POLY.OBJECTID AS PREFECTURE_ID FROM SPATIAL_POINTS AS POINTS, JAPAN_POLYGONS AS POLY WHERE ST_INTERSECTS(POINTS.SHAPE, POLY.SHAPE);
    ```

- **Distance Queries**
  - generally determine the distance to a landmark, or they can be used as filter criteria to obtain all other points within a given radius
  - ```sql
      SELECT POINTS.LATITUDE, POINTS.LONGITUDE, ST_DISTANCE(SHAPE, ST_POINT(139.745442,35.658661,4269),'kilometer') AS DISTANCE_KM FROM SPATIAL_POINTS AS POINTS;
    ```

- **Nearest Neighbour Queries**
  - used to determine the top N nearest neighbours. This would be a self join of the point locations (latitude, longitude), eliminating the comparison of the same points and ranking the distance measure to obtain the nearest neighbours
    - ```sql
      WITH DISTANCE_RANKING AS ( SELECT POINTS.LATITUDE, POINTS.LONGITUDE, POINTS_COMP.LATITUDE, POINTS_COMP.LONGITUDE, ST_DISTANCE(POINTS.SHAPE, POINTS_COMP.SHAPE,'mile') AS DISTANCE_MILES, ROW_NUMBER() OVER(PARTITION BY POINTS.OBJECTID ORDER BY DISTANCE_MILES) AS RANKING FROM SPATIAL_POINTS AS POINTS, SPATIAL_POINTS AS POINTS_COMP WHERE POINTS.OBJECTID<>POINTS_COMP.OBJECTID )
      SELECT * FROM DISTANCE_RANKING WHERE RANKING<6
    ```
Spatial Function Types

- Geometric information functions, which return information about a geometric object
- Conversion (or constructor) functions, which convert an object into another representation
- Comparison functions, which evaluate whether two or more objects touch, overlap, or otherwise intersect or connect
- Geometric object manipulation functions, which can set coordinate values or derive new objects such as centroids, buffers, bounding regions, and so on
- Distance and area functions, which evaluate objects for measurements such as area, distance, and length