Tips and Tricks on Spatial Data, SQL Access and Working with SQL Spatial

AFREEN VIRANI
JONATHAN FARMER
Assumptions

Target Audience

- Intermediate knowledge of SQL and relational databases.

- No knowledge of the ST_Geometry data type or functionality is necessary.

- Not covering setup and configuration of ST_Geometry environments.
  - Please stop by the support island.

- Questions at the end of the presentation.

*Please turn off cell phones*
Agenda

• What is ST_Geometry?

• Why use ST_Geometry?
  - Benefits of ST_Geometry

• Additional consideration

• DEMO - How to use ST_Geometry?
  - Constructor
  - Accessors
  - Operators and Relationship
What Is ST_Geometry?

- ST_Geometry is a spatial type that stores geometry data in a single spatial attribute

Spatial Index

Relational and geometry operators and Functions
- Constructors
- Accessors
- Relationship and Operators

<table>
<thead>
<tr>
<th>OBJECTID</th>
<th>SHAPE</th>
<th>PROPERTY</th>
<th>Res</th>
<th>Zoning</th>
<th>SHAPE_L</th>
<th>SHAPE_A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Polygon</td>
<td>5001</td>
<td>Non-Residential</td>
<td>&lt;Null&gt;</td>
<td>3592.7500813</td>
<td>11255.2410839</td>
</tr>
<tr>
<td>2</td>
<td>Polygon</td>
<td>5002</td>
<td>Non-Residential</td>
<td>&lt;Null&gt;</td>
<td>314.9558377</td>
<td>16406.2417709</td>
</tr>
<tr>
<td>3</td>
<td>Polygon</td>
<td>1003</td>
<td>Residential</td>
<td>Residential</td>
<td>450.955523</td>
<td>12816.2913793</td>
</tr>
<tr>
<td>4</td>
<td>Polygon</td>
<td>1004</td>
<td>Residential</td>
<td>Residential</td>
<td>5217.751249</td>
<td>14036.353453</td>
</tr>
<tr>
<td>5</td>
<td>Polygon</td>
<td>1005</td>
<td>Residential</td>
<td>Residential</td>
<td>452.479849</td>
<td>6816.352685</td>
</tr>
</tbody>
</table>
Why use ST_Geometry?

Benefits of ST_Geometry

- **Enhances Efficiency**

- **Interact with data on the SQL level**
  - Create tables, with a spatial attribute
  - Read and analyze the spatial data
  - Insert, update and delete *simple* features

- **Accessed using common API’s and SQL**
  - International Organization for Standards (ISO) compliant
  - Open Geospatial Consortium, Inc. (OGC) compliant.

- **Bridge the gap between GIS and non-GIS users**

- **Sometimes you want a single result, and not a map**
Editing Geodatabase Feature Classes using SQL

Additional considerations

When working outside of ArcGIS, keep in mind:

• What can you edit?
  - Simple features (points, lines, polygons)
  - Without geodatabase behavior (*Is_Simple*)

• Editing Versioned Tables (*versioned view*)

• Must maintain next ObjectID and GlobalID values (*Next_RowID/Next_GlobalID*)

• Minimal validation of the objects will be performed
Rules for creating spatial tables to be used with ArcGIS

Prerequisites

- Unique identifier.
- One spatial column in the table.
- One spatial reference in the table.
- Do not use mixed-case object names.
- Entity type matches the type defined for the spatial column.
ST_Geometry Functions

Demos

Relational and geometry operators and Functions

- **Constructors** – creates new geometry
  - Example: ST_Point, ST_Line, ST_Polygon

- **Accessor** – return property of a geometry
  - Example: ST_Area, ST_SRID

- **Relationship and Operators** – perform spatial operations
  - Example: ST_Intersects, ST_Buffer
1. Constructors
2. Accessors
3. Relationship and Operators
Additional Resources

• ST_Geometry Function List

• Configuring ST_Geometry for SQL Access
  - Oracle
  - PostgreSQL

• Spatially enable an SQLite database
Please Take Our Survey on the **Esri Events App**!

Download the Esri Events app and find your event

Select the session you attended

Scroll down to find the survey

Complete Answers and Select “Submit”