SQL access and working with ST_Geometry Functions

Christian Wells & Ken Galliher

Right place?

Target Audience

- Intermediate understanding of SQL syntax.
  - Fundamental DML and DDL concepts.
  - Executing functions in SQL.
- No knowledge of the ST_Geometry data type or functionality necessary.
- Not covering setup and configuration of ST_Geometry environments.
  - Please stop by the support island.

Agenda

ST_Geometry

CONSTRUCTORS

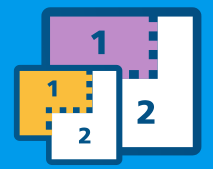
What is ST_Geometry

ACCESSORS

Why ST_Geometry

OPERATORS

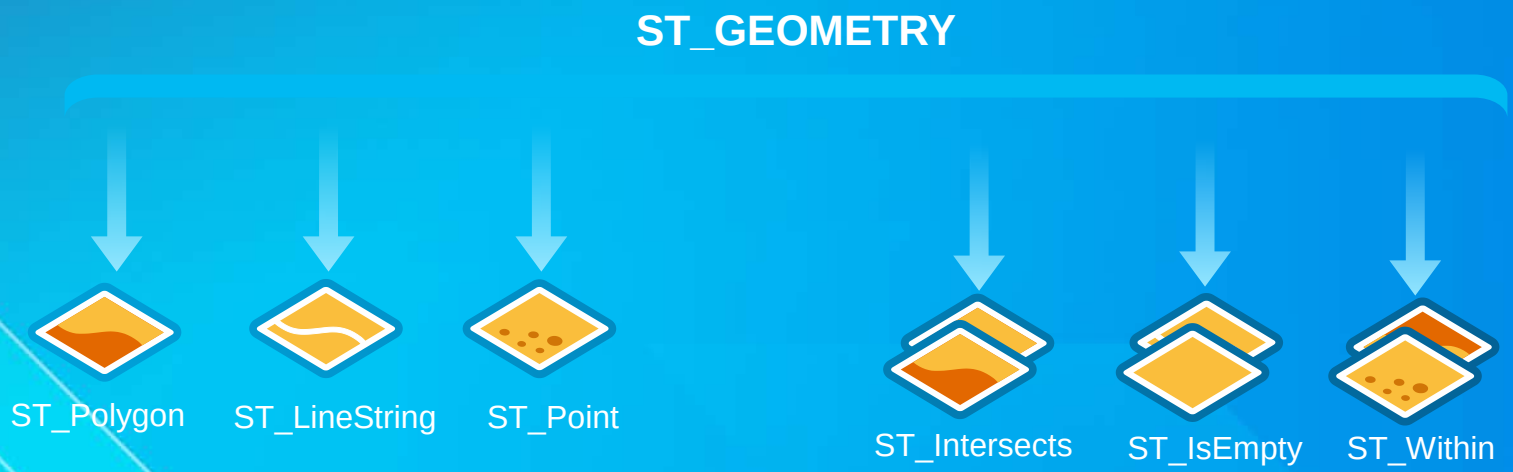
RELATIONSHIP



What Is ST_Geometry?

ST_Geometry

- User defined data type to store spatial data.
- Allows for SQL access to spatial geodatabase and database objects.
- International Organization for Standards (ISO) and Open Geospatial Consortium, Inc. (OGC) compliant.



Why ST_Geometry?

- Possible to use SQL DML and DDL to:
  - Access/Modify geodatabase data via SQL.
  - Integrate spatial data with business data.

- Use Cases:
  - Bulk Editing.
  - Reporting Applications.
  - Non-visual Spatial Queries
  - Truncate/Append

How is ST_Geometry implemented?

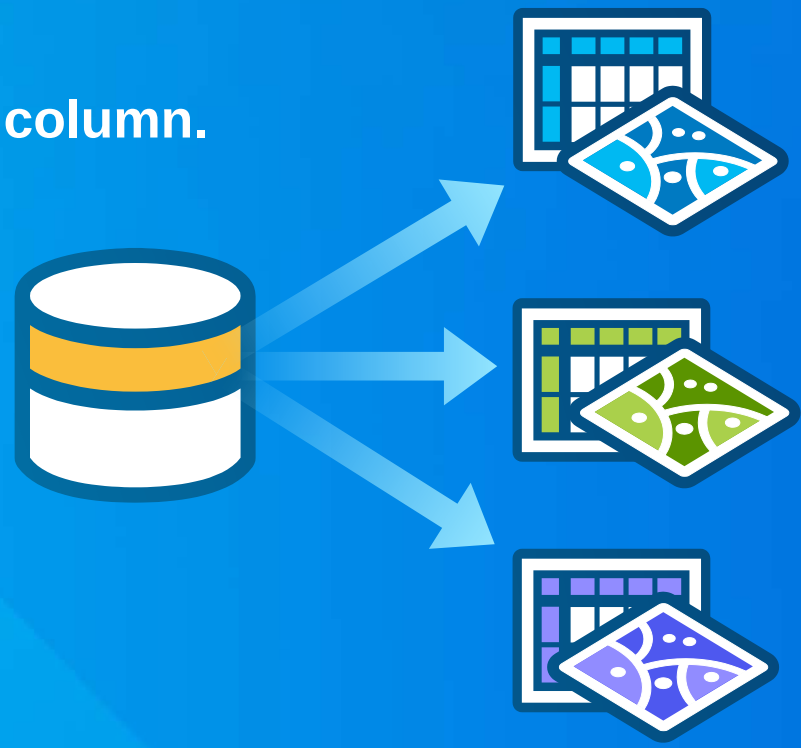
Implementing the data type

- Oracle & PostgreSQL
  - ST_Geometry is the default geodatabase geometry storage type.
  - Create Spatial Type geoprocessing tool. – install/upgrade
- SQLite
  - Create SQLlite geodatabase with ST_Geometry for use in ArcGIS applications.
  - CreateSQLiteDatabase geoprocessing tool.
- IBM
  - DB2 & Informix - Spatial Extender and Spatial DataBlade only.
- SQL Server
  - ST_Geometry is not available. Use Geometry or Geography type.

Rules for creating spatial tables to be used with ArcGIS

Prerequisites

- Unique, non-null column to be used as a unique identifier.
- Avoid multiple spatial columns in the same table.
- Do not use mixed-case object names.
- Entity type matches the type defined for the spatial column.
- One spatial reference on the spatial column.



DEMO

Creating ST_Geometry Tables

Editing Feature Classes with SQL

Additional considerations

- **When working outside of ArcGIS, keep in mind:**
  - **Minimal validation of geometry objects will be performed**
  - **Must not be part of complex geodatabase objects (**Is_Simple**)**
  - **Must maintain next ObjectID and GlobalID values (**Next_RowID/Next_GlobalID**)**
  - **Editing versioned feature classes through its versioned view (**Versioned View**)**

ST_Geometry: Functions

Constructors:

- Create a geometry from:
  - well-known text (WKT)
  - well-known binary (WKB)
  - shapefiles (PostgreSQL Only).
- Data inserted is defined either as point, linestring, polygon, or their variants:
  - `sde.st_geometry('POINT (1 2)', 4326)`
  - `sde.st_geometry('LINESTRING (33 2, 34 3, 35 6)', 4326)`
  - `sde.st_geometry('POLYGON ((3 3, 4 6, 5 3, 3 3))', 4326)`
- The Optimized point constructor can be used:

```
sde.st_geometry(x, y, z, m, srid)
```

ST_Geometry: Functions

Accessors:

- Return the properties of a geometry:
  - Dimensionality: `ST_Dimension`
  - Coordinates: `ST_Is3D, ST_IsMeasured, ST_X, ST_Y, ST_Z, ST_M`
  - Envelope: `ST_Envelope`
  - Geometry Type: `ST_Geometry, ST_Entity`
  - Simple: `ST_IsSimple`
  - Empty: `ST_IsEmpty`
  - Closed: `ST_IsClosed`
  - SRID: `ST_SRID`

DEMO

Constructors & Accessors

ST_Geometry: Functions

Relational Operators

Relationships:

- Test for different types of spatial relationships with predicates.
- Compares relationships:
  - Exteriors of geometries
  - Interiors of geometries
  - Boundary of geometries
- Predicates compare the x- and y-coordinates of the submitted geometries.
- Sample Functions: `ST_Contains, ST_Intersects, ST_Within, ST_Touches`

ST_Geometry: Functions

Functional Operators

Operations:

- **Take spatial data to produce a new output.**

- **Buffer:** `ST_Buffer`
- **Union:** `ST_Union`
- **Difference:** `ST_Difference`
- **Intersection:** `ST_Intersection`

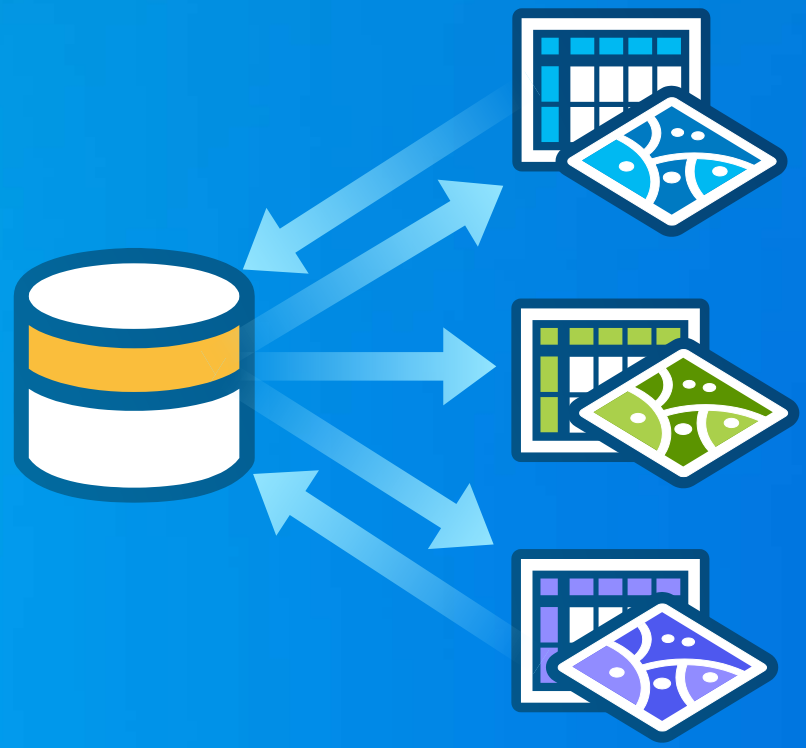
- `sde.ST_Intersects(a.shape, sde.ST_Buffer(b.shape, 0.5))`

DEMO

Relationships & Operations

ST_Geometry Performance

- Things to consider:
  - More rows = more time
  - Row by row comparison
- Versioned Views
  - No Domain Index (Oracle)
  - Complete Materialized View refresh
- Drop spatial index before bulk data loads.
- Transport rows as binary.



ST_Geometry: Validation

- INSERTS and UPDATEs through ArcGIS - ST_Geometry validation occurs automatically.
- Some validation is performed inserts, including SRID and construction rules.
  - Construction Rule Examples:
    - Polygons must close
    - Lines must have at least 2 points
- Use ST functions to check for construction rule violations.
  - ST_IsEmpty, ST_IsClosed, IsRing, etc.
- Check the ST_Geometry byte stream for issues.

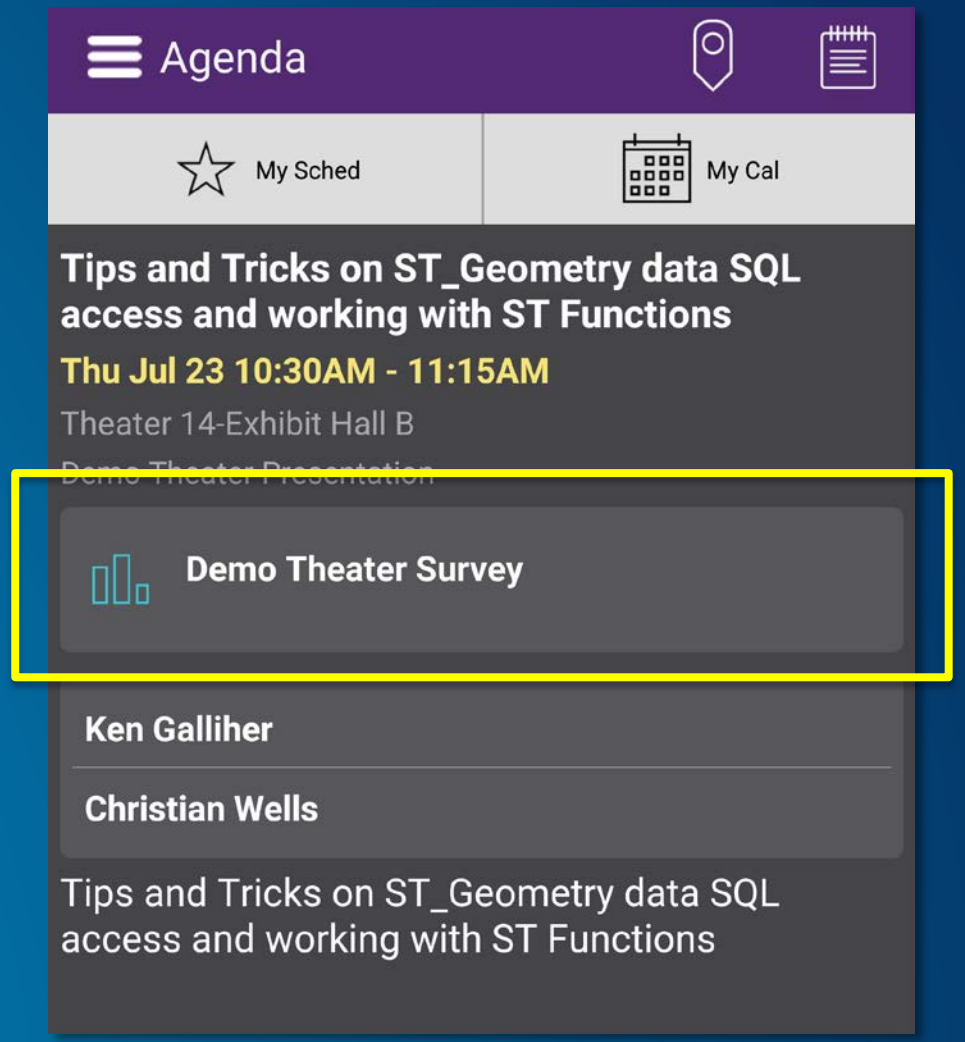
DEMO

Validation & Performance

Questions

Thank you!

- Please fill out the session survey in the mobile app.
- Select Tips and Tricks on ST_Geometry data SQL access and working with ST Function.
- Click on “Demo Theater Survey”
- Answer a few short questions and leave comments.



Want to learn more?

- **Configuring ST_Geometry for SQL Access:**
  - **Oracle**
  - http://desktop.arcgis.com/en/desktop/latest/manage-data/gdbs-in-oracle/configure-oracle-extproc.htm
  - **PostgreSQL**
  - http://desktop.arcgis.com/en/desktop/latest/manage-data/databases/add-the-st-geometry-type-to-a-postgresql-database.htm
- **ST_Geometry Function List**
  - http://desktop.arcgis.com/en/desktop/latest/manage-data/using-sql-with-gdbs/st-geometry.htm
- **Spatially enable an SQLite database:**
  - http://desktop.arcgis.com/en/desktop/latest/manage-data/databases/spatially-enable-sqlilte.htm

ESRI Technical Support 888-377-4575

Understanding our world.