Migration from Other Spatial Formats to ST_Geometry in the Geodatabase

Ben Lin & Damion Scholz
Agenda for this presentation

Overview

• What is ST_Geometry?
• ST_Geometry performance
• Why migrate to ST_Geometry?
• The benefits of using ST_Geometry
• Query layers with ST_Geometry
• ST_Geometry functions
Migration from Other Spatial Formats to ST_Geometry in the Geodatabase
• This storage type extends the capabilities of the database.
ST_Geometry Performance

• Make efficient use of database resources.

• Compatible with the database.

• Provide rapid access to spatial data.
Why consider migrating to ST_Geometry?

There are two reasons you would migrate your data (Oracle):

• To access your spatial data.

• To move from an older data type.
Which data types can migrate to ST_Geometry?

- SDO_Geometry
- SDELOB
- ST_GEOMETRY
Tips and Tricks

• Version views and spatial views are recreated automatically after migration to ST_Geometry

• Database views will be recreated after migrating to ST_Geometry. If the view contains a spatial column, it will convert to a spatial view.

• Once migrated to ST_Geometry, data cannot be reverted.
Benefits of using ST_Geometry

```sql
SQL> select CONFIG_STRING from dbtune
where keyword = 'DEFAULTS' and parameter_name = 'GEOMETRY_STORAGE';

CONFIG_STRING
_________________________________________
ST_GEOMETRY
```
Use Query Layer for ST_Geometry data

- Can be used to display only features that meet a specific criteria.
Types of SQL functions

- **Constructor Functions**
- **Accessor Functions**
- **Relational Functions**
- **Geometry Functions**

T/F
Constructor Function: **ST_Geometry**

constructs a geometry from a well-known text representation.

**Syntax:**
```
Sde.st_geometry (wkt clob, srid integer)
```

**Example:**
```
SQL> INSERT INTO ucdemo.trees (objectid, tree_age, shape) VALUES (1902, 14, sde.st_geometry ('point (14 7)', 4326));
1 row created.
```
Types of SQL functions

- Constructor Functions
- Accessor Functions
- Relational Functions
- Geometry Functions

T/F
Accessor Function: **ST_isSimple**

ST_IsSimple returns 1 (Oracle and SQLite) or t (PostgreSQL) if the geometry object is simple as defined by the OGC; otherwise, it returns 0 (Oracle and SQLite) or f (PostgreSQL).

**Syntax:**

```
Sde.st_isSimple (geometry1 sde.st_geometry)
```

**Example:**

```
SQL> SELECT objectid, sde.st_isSimple (shape) Is_it_simple
       FROM ucdemo.trees where objectid = 1902;

OBJECTID IS_IT_SIMPLE
---------- -----------
1902        1
```
Types of SQL functions

- Constructor Functions
- Accessor Functions

Relational Functions

Geometry Functions

T/F
DEMO:
Oracle 12c, ArcSDE 10.2.2 Geodatabase

Migrate Storage Geoprocessing tool,
Functions:
ST_Buffer, ST_Intersect

Ben Lin & Damion Scholz
Thank you…

- Please fill out the session survey:

First Offering ID: 1275
Second Offering ID: 1901

Online – www.esri.com/ucsessionsurveys
Paper – pick up and put in drop box