Accessing your Enterprise Geodatabase using SQL

Amit Kesarwani & Said Parirokh
Assumptions

- Intermediate knowledge of SQL and relational databases
- Intermediate knowledge of the geodatabase
- We’ll hold all questions until the end

Please turn off cell phones
Badly Behaving Clients

- Incorrect Editing Patterns
Agenda

- RDBMS support in ArcGIS
- Geodatabase and it’s system tables
- Demo – Querying the geodatabase schema
- Leveraging spatial functions
- Demo – Spatial analysis using SQL
- Editing geodatabase feature classes using SQL
- Discovery functions
- Demo – Editing versioned and non-versioned feature classes
- Summary of guidelines
ArcGIS Supports Multiple Implementation Patterns

Database Centric

Server Centric

Web Centric
Accessing data in a DBMS

You can access spatial or non-spatial data in a DBMS to use in ArcGIS

- **Geodatabase**
  - Supported on:
    - DB2
    - Informix
    - Oracle
    - PostgreSQL
    - SQL Server
    - SAP HANA

- **Database - Simple feature access**
  - Supported on:
    - ALTIBASE
    - Dameng
    - Teradata
    - Netezza
    - DB2, Informix, Oracle, PostgreSQL, SQL Server, SAP HANA
Geodatabase - built on top of database functionality

- When you want to do more with your data
- Control data integrity - domains and subtypes
- Advanced data modeling - transportation or utility networks
- Store and work with detailed cartography
- Multi-user editing – high isolation
Geodatabase schema

- There are two sets of tables
  - Dataset tables (user-defined)
  - Geodatabase system tables (schema is controlled by ArcGIS)
What’s in a Geodatabase

- Look at Geodatabase structure
Geodatabase system tables

- Track contents within a geodatabase
- Store definitions, rules and behavior for datasets
- Geodatabase schema is stored primarily within an XML field
- Query contents only
Geodatabase schema – four main system tables

**GDB_Items**
- List all geodatabase items

**GDB_ItemTypes**
- Fixed list of items

**GDB_ItemRelationships**
- Lists all relationships

**GDB_ItemRelationshipTypes**
- Fixed list of relationships
Geodatabase schema

Definition field

- XML document for each item
  - Native XML:
    - SQL Server
    - PostgreSQL
    - DB2
  - ArcSDE XML:
    - Oracle
    - Informix
Accessing your geodatabase using SQL

• Query schema and properties of existing datasets
  - Use SQL statements to query the definition field on the gdb_items table
    - Oracle and Informix use ArcSDE XML
    - Gdb_Items_vw / gdb_itemrelationships_vw

• Editing tables/feature classes, whether versioned or not
  - Use views with versioned/archive classes

• Create tables with SQL that contain spatial or raster types

• Leverage SQL functions to evaluate attributes and spatial relationships, perform spatial operations, and return and set spatial properties.
Accessing your geodatabase using SQL

- With SQL, you access the data at the DBMS level
  - Bypass behaviors and functionality enforced by the geodatabase or ArcGIS clients

- Need to be aware of what you can and cannot edit
  - Know your data
  - Use discovery functions
Querying the geodatabase schema

- Look at system tables
- Query geodatabase contents
  - Feature classes within a Feature dataset
  - List domains with their code/values
  - Find datasets that are using a domain
Querying geodatabase user-data

- Why use SQL when I have a GIS?
  - Use power of dbms engine to query and analyze your data
  - DBMS spatial methods for performing spatial analysis
  - Bridge between GIS and Business Intelligence / Insights
  - Sometimes you want a single result, and not a map
What is a spatial type?

- A type that stores geometry data in a single spatial attribute
  - Geometry type, coordinates, dimension, spatial reference

- Spatial index
  - Improves spatial searches

Relational and geometry operators and functions

- Constructors – creates new geometry
- Accessor – return property of a geometry
- Relational – perform spatial operations
- Geometry – transform from one geometry to another
Benefits of a spatial type

- With SQL and a spatial type you can
  - Create tables with a spatial attribute
  - Read and analyze spatial data
  - Insert, update and delete *simple* features

- Enhances efficiency
  - Data and methods are stored in the database
  - Applications access native dbms type

- Access using common APIs and SQL
  - Standard functions
  - Well-known interchange formats
Viewing database data in ArcGIS

- Simple SQL query

```sql
SELECT OBJECTID, SHAPE
FROM GisUser.INCIDENTS
```
Viewing database data in ArcGIS

```sql
WITH neighborhoodIncidents AS
(
    -- Filtered Incidents in the neighborhood
    SELECT
        i.OBJECTID,
        i.SHARE,
        i.INCIDENT_DATE,
        Neighborhood = n.Neighborhood
    FROM
        GisUser.INCIDENTS i
    JOIN
        GisUser.NEIGHBORHOODS n
    ON
        i.Shape.STIntersects(n.Shape) = 1
    WHERE
        i.CATEGORY = 'BURGLARY'
        OR
        i.CATEGORY = 'GRAND THEFT AUTO'

    -- Predictive Analysis Of High Risk Incident Zones
    SELECT
        n.OBJECTID,
        n.SHAPE,
        n.Inc.NeighborCount,
        n.Inc.AvgDateDifference,
        n.Neighborhood,
        n.INCIDENT_DATE
    FROM
        (-- Incidents with neighbors
            SELECT
                n.OBJECTID,
                NeighborCount = COUNT(i.OBJECTID),
                AvgDateDifference = ABS(AVG(DATEDIFF(WEEK, n.INCIDENT_DATE, i.INCIDENT_DATE)))
            FROM
                neighborhoodIncidents n
            JOIN
                GisUser.INCIDENTS i
            ON
                n.Shape.STBuffer(500).STIntersects(i.Shape) = 1
            GROUP BY
                n.OBJECTID
        ) nInc
    JOIN
        neighborhoodIncidents nInc
    ON
        nInc.OBJECTID = n.OBJECTID
)
```
Spatial Analysis using SQL

- Querying geodatabase data
- Spatial methods
Creating geodatabase feature classes using SQL

- Use SQL to create and populate tables

```sql
CREATE TABLE hazardous_sites
    (oid INTEGER NOT NULL, site_id INTEGER,
     name VARCHAR(40), location sde.st_geometry)
```

- Need to register the table with the geodatabase to participate in geodatabase functionality.
Editing geodatabase feature classes using SQL

• What can you edit?
  - Simple features (points, lines, polygons)
  - Without geodatabase behavior
  - Use the `Is_Simple` function to determine whether your data can be edited

• Editing non-versioned tables
  - Edit tables directly

• Editing versioned tables
  - Edit special `versioned view` instead of tables
The ObjectID field

- Every geodatabase feature class or table has an ObjectID field
  - Unique, not null integer
- Value is generated and maintained by ArcGIS
- Non-versioned inserts
  - Use `RowID_Name` to determine name of ObjectID field
  - Use `Next_RowID` function to obtain next ObjectID value
- Versioned inserts
  - ObjectID obtained automatically by versioned view
- *Never* update an ObjectID field
Editing versioned tables and feature classes

• Versioning
  - Supports concurrent editing with long transactions
  - Undo/redo experience
  - No locking or data extraction required

• All changes written to delta tables
  - Adds (a) table and Deletes (d) table
  
• Edits are assigned an identifier (state_id)
  
• A version references a lineage of state_ids
Versioned views

- Specialized view to work with versioned data using SQL
  - Uses stored procedures, triggers and functions to access and edit versioned tables
- Result set is based on versioned query
- Created on a single versioned table, contains all columns
- Created automatically when feature class or table is versioned
- Versioned must be reconciled through ArcGIS
Discovery Functions

- **Is_simple** – returns false/true or not registered with Geodatabase
- **Is_Versioned** - returns false/true or not registered with Geodatabase
- **Version_View_Name** - returns the name of the versioned view defined on that table
- **Is_Archive_Enabled** - determines if the specified table has been enabled for archiving.
- **Archive_View_Name** – returns name of the archive view on a nonversioned, archive-enabled table
- **Is_Replicated** - returns false/true
- **Geometry_Columns** - returns the name of all spatial columns in the table that is not registered with Geodatabase
- **GlobalID_Name** - returns the name of the global ID field in the table.
- **RowID_Name** returns the name of a table's ObjectID (or RowID) field
Editing

- Discovery functions
- Non-versioned edits
- Versioned view edits
Guidelines for using SQL and the geodatabase

- Understand the geodatabase system tables and their structure
- Avoid changing data that affects geodatabase software level behavior
- Geodatabase awareness
  - You have it
  - The database does not
Guidelines for using SQL and the geodatabase

<table>
<thead>
<tr>
<th></th>
<th>GDB System tables</th>
<th>Simple FC / Tables</th>
<th>Complex FC / Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERY</td>
<td>✓</td>
<td>✓</td>
<td>No</td>
</tr>
<tr>
<td>Edit/Update</td>
<td>☠️</td>
<td>✓</td>
<td>☠️</td>
</tr>
<tr>
<td>Insert</td>
<td>☠️</td>
<td>✓</td>
<td>☠️</td>
</tr>
</tbody>
</table>
Guidelines for using SQL and the geodatabase

- Do perform spatial operations
- Do query spatial and attribute information
- Do INSERT, UPDATE and DELETE geometries
  - As long as you pay attention to behavior
- Do INSERT, UPDATE and DELETE attribute data
  - As long as you pay attention to behavior
- Do write efficient SQL
Guidelines for using SQL and the geodatabase

• DO NOT update the objectid (row_id) value
• DO NOT modify geometries for feature classes participating in
  - Topologies, geometric networks, network datasets, terrains, parcel fabrics, geodatabase replication, schematic datasets, feature-linked annotation…
• DO NOT update attributes that define geodatabase behavior
  - Enable/Disabled attributes, ancillary attributes, weight attributes…
• Use `is_Simple` to check
Resources

• Comprehensive documentation covering
  - Accessing dataset properties
  - Editing geodatabase data
  - Esri spatial and raster type reference

• Get started at:
Online survey

- Please give us your feedback in the Esri Events app

Accessing your Enterprise Geodatabase using SQL
http://www.esri.com/events/eventsapp