Designing and using a Geodatabase

Gudmundur Hafberg, Andy MacDonald, Tom Brown,
Goals

- Understanding of:
  - Basic geodatabase data model design
  - The importance of a well tuned database
  - Data loading techniques
  - Effective use of ArcMap and ArcCatalog
  - Troubleshooting
Data model design
The geodatabase provides many powerful new concepts:
- geometric networks
- relationship classes
- validation rules

Proper modeling is critical to good performance
Feature datasets

• Container for feature classes
  – shared spatial reference

• Analogous to a coverage
  – less restrictive

• May also contain
  – relationship classes
  – geometric networks
Feature datasets

- Scope for a spatial reference
- Scope for topology
- Opening a feature dataset containing lots of feature classes is slow the first time (ArcInfo 8.0.x)
  - opening a single feature class results in the feature dataset being opened
- Don’t overload feature datasets
  - only group feature classes that you work with at the same time in a feature dataset
Feature dataset

![Feature Dataset Properties](image)
Geometric networks

• Connectivity relationship between network feature classes
• Used to model network systems
• Network connectivity:
  – based on geometric coincidence
  – always live
• Feature classes in the same feature dataset
Network topology
Geometric networks

• Topology maintained on the fly
  – connectivity based on geometric coincidence
  – when adding a new feature, all other feature classes are searched

• Minimize the number of network feature classes
  – utilize subtypes
Subtypes

• Different types of features or objects in an object class
  – same attributes
  – same behavior
  – can have different validation rules
    • attribute domains
    • default values
    • connectivity rules
    • relationship rules
• Defined by the subtype code field value
Subtypes

- In a feature class which stores pipes
  - steel pipes can be between 6 and 36 inches in diameter
  - concrete pipes can be between 24 and 240 inches in diameter

- Steel and concrete are subtypes of the pipe feature class
  - diameter attribute domain differs
Subtypes
Geometric networks

• When minimizing network feature classes
  – *consider* fat classes
    • handle unpopulated attributes
  – *consider* denormalizing and subtyping
    • cache the necessary attributes
    • requires custom features
Geometric networks

• Subtyping caveats
  – may require custom features
  – cannot snap to subtypes
  – cannot cloak fields
  – can be costly in certain circumstances (e.g. layer definition queries)
Geometric networks

• When editing networks *always* use the Edit Cache
  – reduces the number of spatial queries against the server when discovering connectivity

(Edit cache stores select features in local memory – needs to be rebuilt when the AOI changes)
Relationships

• An association between objects
  – feature to row, feature to feature...

• Stored in a relationship class

• Related objects can message each other
  – origin to destination, destination to origin, both, neither
  – can trigger behavior (cascade delete, move to follow, custom...)
Relationships

Composite relationship, Pole to Transformer

Select a pole and move it …the transformer follows
Relationship classes

• Relationships link objects
  – updates trigger notification
  – composites trigger behavior
  – navigation more expensive than INFO relates

• Feature-linked annotation is maintained through composite relationships
F-linked annos (example)
Relationship classes

- Index primary and foreign keys
- Add all related classes to the map
  - open/close cycle...
- Try to symbolize based on attributes in the feature class (joins are expensive!)
- Update on the source class can trigger update on the target class (ex. F-linked annos)
Database tuning
Tuning

• Poorly tuned DBMS results in a poorly performing geodatabase
• Follow RDBMS and ArcSDE tuning guides
  – configure the RDBMS to reduce disk contention
    • control, redo log and archive files
    • system and user tablespaces, etc
I/O

Database

System files

(1) ORACLE_HOME
SDEHOME
Control file 1
SOE

(2) Redo logfile 1
Redo logfile 2

(3) Redo logfile 3
Redo logfile 4

(4) Control file 2
SYSTEM

(5) ROLLBACKS1
(segments 01, 03, 05, 07, 09)

(6) ROLLBACKS2
(segments 02, 04, 06, 08, 10)

Application data

(7) USER'S TABLESPACE
Archive destination

(8) USER_TEMP

(9) BUSINESS TABLES

(10) FEATURE TABLES

(11) SPATIAL TABLES

(12) INDEXES

(13) DELTA TABLES

(14) NETWORK TOPOLOGY TABLES

Software installation
File
Disk device
Tablespace

Twentieth Annual ESRI International User Conference • June 26-30, 2000
Tuning (cont)

• Allocate “enough” memory on the server for your database
• Use servers with multiple CPU’s
• Physical network – there is a lot of client/server traffic ongoing → the throughput of the network is critical to the performance of the client application.
Hardcore DBMS tuning

• Sessions
  – Tuning and configuring ArcSDE for Oracle SQL server and Informix
  – Administering a multi-versioned ArcSDE database
Data loading
Data loading

• Try to load all data
  – before building networks
  – before versioning the data

• Pre-process the data using SQL before versioning
  – simple attribute updates
  – linking imported coverage annotation
Data loading

- If large data loads are required after the database is versioned
  - run compress to get new features into the base table
Data loading – Strategy 1

- Schema Generation Wizard to create empty geodatabase schema
- Delete any networks
- Simple Data Loader to load data into simple feature classes
- Build networks
- Reapply Schema Generation Wizard
  - connectivity rules and class extensions
- Register data as versioned
Data loading – Strategy 1

• **Advantages**
  – fast – no network connectivity
  – no versioning impact

• **Disadvantages**
  – custom creation behavior not executed
  – feature-linked annotation
    • generate after loading
Data loading – Strategy 2

- Schema Generation Wizard to create empty geodatabase schema
- Simple Data Loader to load data into simple feature classes
- Register data as versioned
- Object Loader to load data into network classes
- Run compress
Data loading – Strategy 2

- **Advantages**
  - executes all custom feature creation behavior

- **Disadvantages**
  - slow – impractical for large numbers of network features (no edit cache)
  - versioning (data in delta tables, requires compress)
Loading annotation

- Converting labels to annotation
- Converting coverage annotation
- Convert your annotation before versioning your data
  - delta tables, compress
- Convert your annotation before building networks
  - feature snapping does not message annotation
Pre-processing using SQL

• Do not update records in SQL after the data is versioned
• Do not modify attributes that trigger behavior in other objects

• *Never* update the following fields
  – OBJECTID
  – Enabled or AncillaryRole (logical network will not be updated)
  – Network weight fields
Compressing your database

- Performance can degrade over time as more edits are made to the database
- Compress will remove unreferenced database states and redundant rows
  - improves performance
  - can only be run when no-one is working on the database
Compressing your database

- Compress should be run periodically throughout the lifetime of a database
- To get the most out of compress
  - For each outstanding version
    - reconcile and post against DEFAULT
    - delete the version
  - Run Compress
  - Recreate the versions as needed
Compressing your database

• Update database statistics after running `compress`
  – `sde:table administration command`
  OR
  – `ANALYZE <table> COMPUTE STATISTIC` (Oracle)

• Update DBMS statistics periodically
Bulk appending data
Bulk appending data

• Standard approach
  – use the Object Loader
  – run compress

• This interactively builds network connectivity
  – slow process
    • impractical for large numbers of network features (no edit cache)

• All object behavior is executed
Hardcore bulk appending data
Hardcore bulk appending data

- Faster way to bulk append data
  - heavy workflow impact
  - be careful!
Hardcore bulk appending data

• Follow this sequence
  – reconcile and post all outstanding versions to DEFAULT and delete the versions
  – compress the database
  – unversion the data
  – drop the network
  – load the new data (Simple Data Loader)
  – rebuild the network, reapply Schema Wizard
  – register the data as versioned
  – recreate versions as required
Hardcore bulk appending data

- Limitations with this method
  - can’t handle complex junctions with custom connection points
  - disconnected network features will be re-connected
  - will not execute any object behavior (feature-linked annotation)
Application tuning
Application tuning

• There are dos and don’ts for the effective use of ArcCatalog and ArcMap

• ArcCatalog
  – create thumbnails for browsing the contents of your database
    • no need to open the feature class
    • prevents unnecessary data retrieval
Thumbnails
Application tuning

- ArcMap
  - use scale suppression, especially with annotation
  - always start with a zoomed in view
  - create overview layers for browsing
  - only include necessary classes in the map
  - include all related classes
  - simplify your symbology
  - use the edit cache
ArcMap document

- Has multiple feature layers
- Has to visit every class when opening the document.
- Identify/Select performance relates to the # of classes in the doc and the setting of selectable and identifiable layers
Application tuning

• Drawing annotation is expensive
  – use scale suppression

• Shared editing tools work on all classes in a feature dataset
  – include all classes in the map

• Related classes will be opened on demand
  – include related classes in the map
Application tuning

• Use the edit cache
  – caches features on the client
  – makes network editing faster (reduces the number of spatial queries against the server)
  – caches editable features in the map extent
  – cleared when you stop editing
  – cache must be rebuilt after pan
  – user manages the edit cache
Application tuning

- Different database accounts for different users
  - avoids contention writing to the selection log tables
  - edit cache caches only those features which are being edited
Troubleshooting
Troubleshooting

Look for bottlenecks in your system to improve performance

• Application tuning
• Database design and indexing
  – Look for missing indexes, make sure your tables are analyzed
  – Avoid over-indexing
• Database tuning
  – ArcSde performance = DBMS performance
Patch 8.0.2-10ds

- Get the latest Arc8 patch (both ArcSde and ArcInfo fixes) – Patch 802-10ds for Oracle8i
- Available on ArcOnline
ArcInfo Desktop:
 
ArcMap:

- CQ00108637 - Expose Interface(s) required to convert Coverage Annotation to Geodatabase Annotation programmatically (ArcMap Anno)
- CQ00111000 - Annotation disappearing when moved (ArcMap Anno)
- CQ00111065 - Completely within query is SDE does not select all the features (ArcMap Selection)
- CQ00112805 - normal.mxt being overwritten (ArcMap App Framework)
- CQ00113734 - Need to be able to update annotation feature class extension properties (ArcMap Anno)
- CQ00115275 - 8.0.2 VOFE provide mechanism for changing the size of geodb annotation (ArcMap Anno) Geometry:

- CQ00109455 - ArcMap blows up if you only enter two points while tracking a polygon
- CQ00115707 - Rendering geometries with parametric curves (sde) crashes ArcMap GeoDatabase:

- CQ00112405 - weights not being copied over when merging changes
- CQ00112649 - Fix error chaining for Oracle errors in network building
- CQ00112782 - Undo not working for network weight fields
- CQ00113890 - conflict on a table that is not a FeatureClass, CConflictsDlg::FillList crashes
- CQ00115335 - IWokspaceEdit::StopEditOperation failing under certain conditions
- CQ00116131 - Net builder not pushing weights for junction to the LN correctly
- CQ00116393 - Creating point feature with linked anno - anno not getting populated correctly Network DO:

- CQ00107769 - FindPath causes runtime error when CAD layer in map
- CQ00108521 - ResultEdges causes ArcMap to explode after use Versioning:

- CQ00109000 - map fatals when encountering an unique scenario of a delete/delete conflict

ArcSDE:

- Oracle 8i:

- CQ00108631 - We need to bind all the variables in our version queries to improve the hit ratio (scalability issue)
- CQ00114287 - workaround for Oracle long raw bug TAR 1091134.999 (affects anno and network data) Server:

- CQ00105969 - Server hanging on occasion when making call to license executable
- CQ00111949 - Creating a new feature with f-linked annos on point f-classes does not populate the text attribute
- CQ00113178 - The spatial query for non-versioned gdb annotations fails
- CQ00115222 - Timeout error when connecting to SDE from ArcIMS and ArcMap
- CQ00115984 (CQ00116275 - CQ00116237) - the gsvr crashes when programmatically setting the extent beyond the dataset extent Versioning:

- CQ00114289 - Throw an error when Oracle 1000 element for an IN function is reached. (the fix just prevents corruption due to an Oracle limitation). Performance:

- CQ00114290 - Improve difference query (reconcile performance)
Further Info

• ESRI ArcOnline
  – www.esri.com/usersupport/arconline
• ESRI white papers
  – Multi-user GIS systems with ArcInfo 8
• ArcInfo documentation
  – ArcSDE 8 Tuning Guide
  – Building a Geodatabase
  – Modeling Our World
UC 2K gdb/sde sessions

- Tuning and configuring ArcSDE for Oracle, SQL server & Informix
- Migrating your data to the Geodatabase
- Working with a Versioned Geodatabase
- Administering a multi-versioned ArcSDE Geodatabase
- Geodatabase and Object Model design using CASE tools
- Managing and Editing Geometric networks
- Extending the Geodatabase with class extensions
- Extending the Geodatabase with custom objects
Technical Workshop Survey

• Please fill out the Evaluation Form before you leave!