Enterprise Geospatial Data Management Tips and Tricks

Jim McAbee

jmcabee@esri.com

Enterprise Geodatabase Tips and Tricks

Goals:

…our database is too slow…

- Issues common to all ArcSDE technology consumers.
- Avoid RDBMS specific issues (e.g. Only PostgreSQL)

…I can't add a field to a feature class…

…what is "direct connect" ?...

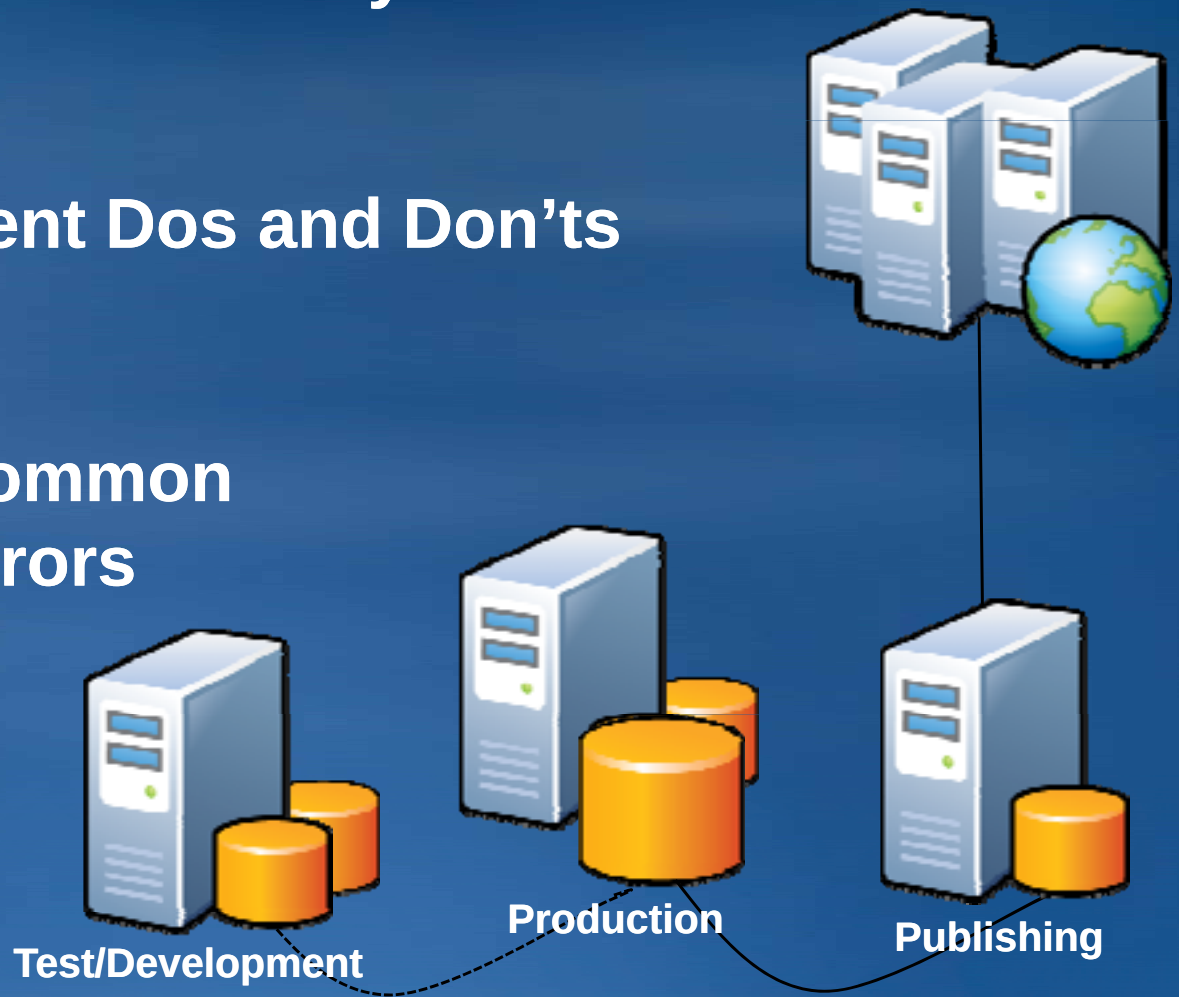
…how do I edit?...

Assumed Knowledge:

- Use of RDBMS tools
- Use of Standard ArcMAP, ArcGIS Server operations

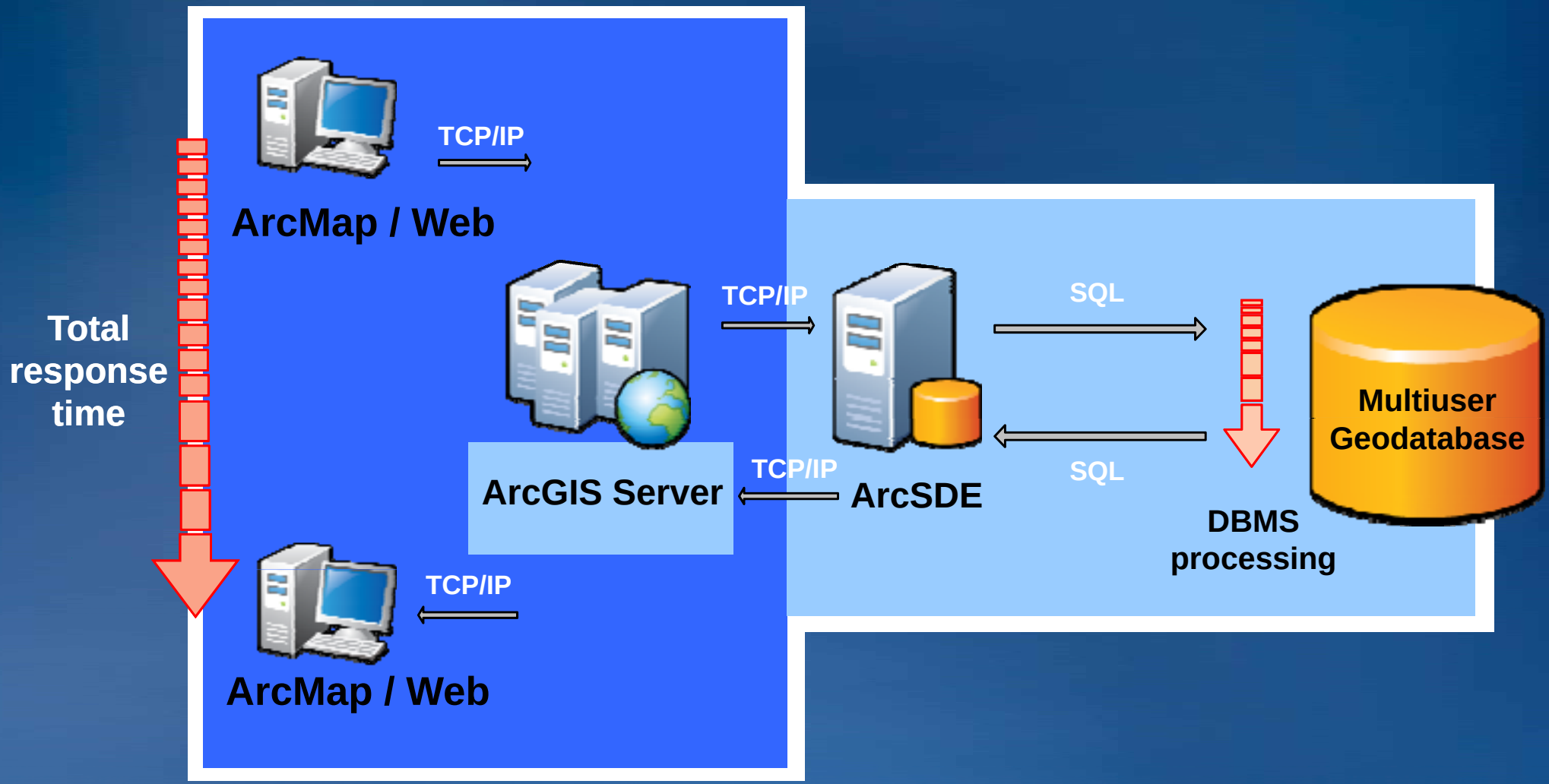
Enterprise Geodatabase Tips and Tricks

- Design
  - Performance and Scalability
- Access
  - Data Management Dos and Don’ts
- Maintenance
  - How to avoid common maintenance errors
- Q&A



Section 1:
Design and Architecture Topics

Design: Performance and Scalability

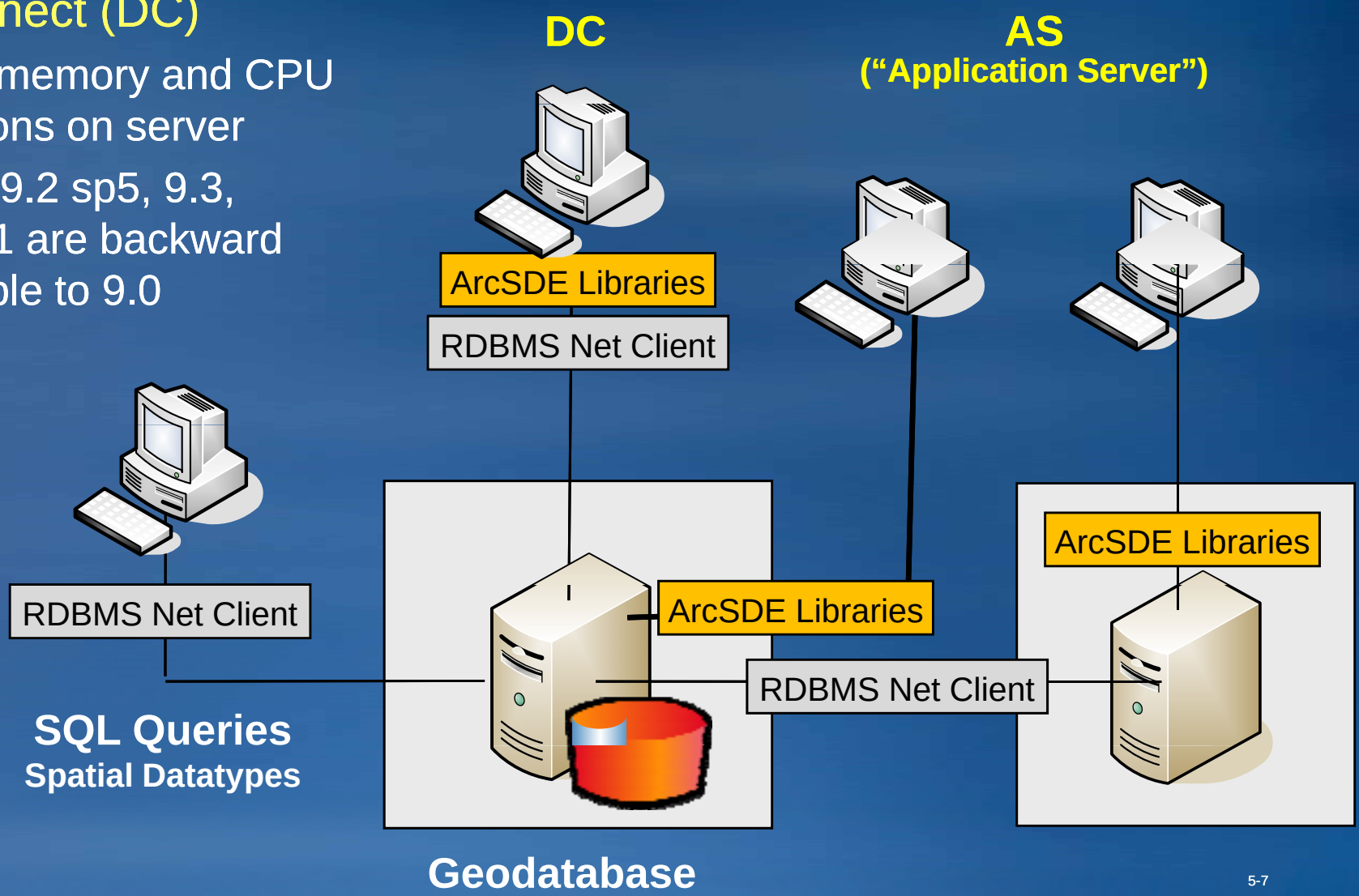


Design: Watch for Bottlenecks

- ArcGIS, Web Browser, and Mobile clients
  - Using proper techniques
- ArcObjects Customization
  - Requires careful design
- Network
  - Easily becomes the bottleneck
- ArcGIS Server
  - Minor tuning required
- Geodatabase and ArcSDE
  - Minor tuning required if any
  - Design an efficient data model
- DBMS
  - Must be tuned for workflow
- Server and Storage Hardware
  - Server O/S must be adequately configured

Design: Geodatabase Connection Architectures

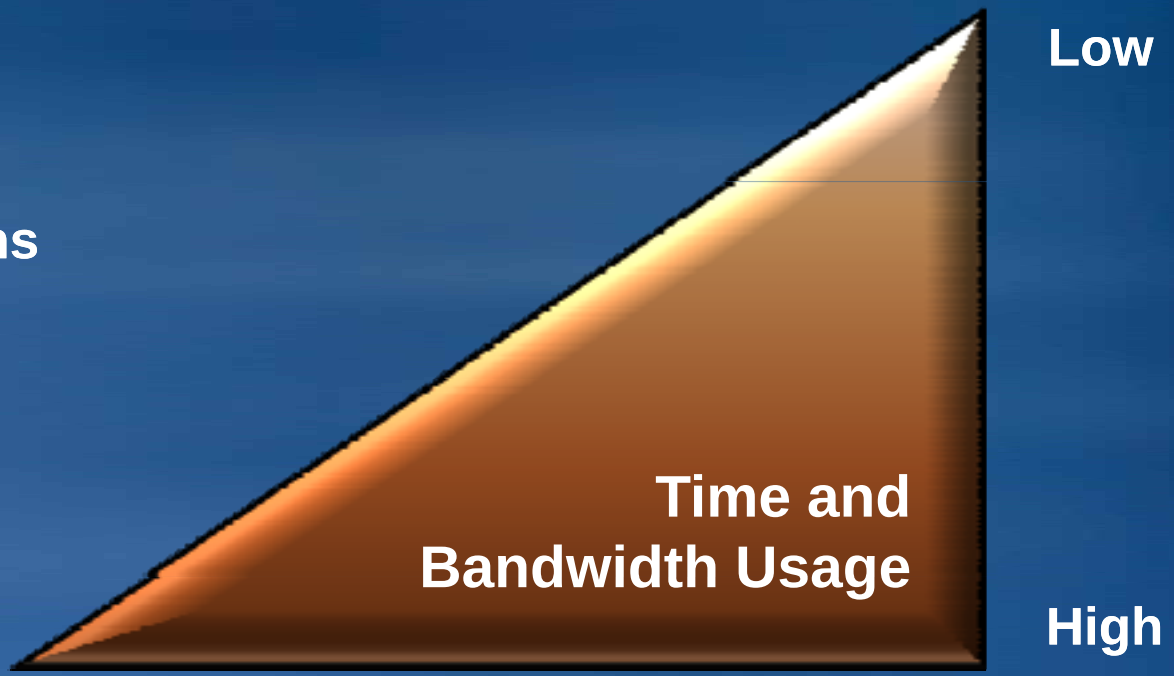
- Application Server Connection (AS)
  - Server memory and / or CPU contention are not issues
  - Light-weight Clients
- Direct Connect (DC)
  - Reduce memory and CPU contentions on server
  - ArcSDE 9.2 sp5, 9.3, and 9.3.1 are backward compatible to 9.0



ArcGIS Connection Network Traffic

- Connection Architectures and Data Sources affect performance and scalability

- Connection Types
  - ArcGIS Server Services
  - Geodatabase connections
    - Direct Connect
    - Application Server
  - File based
    - File Geodatabase
    - UNC file sharing



- Data Type
  - Vector vs. Vector and Raster
- Data Compression
  - Enterprise Geodatabase (Raster) and RDBMS
  - File Geodatabase

ArcSDE initialization parameters

- **SERVER_CONFIG** table records ArcSDE initialization parameters
  - Previous versions used **giomgr.defs** file
- Update using the **sdeconfig** command
  - Dynamic ArcSDE configuration changes
  - Only ArcSDE administrator can alter configuration

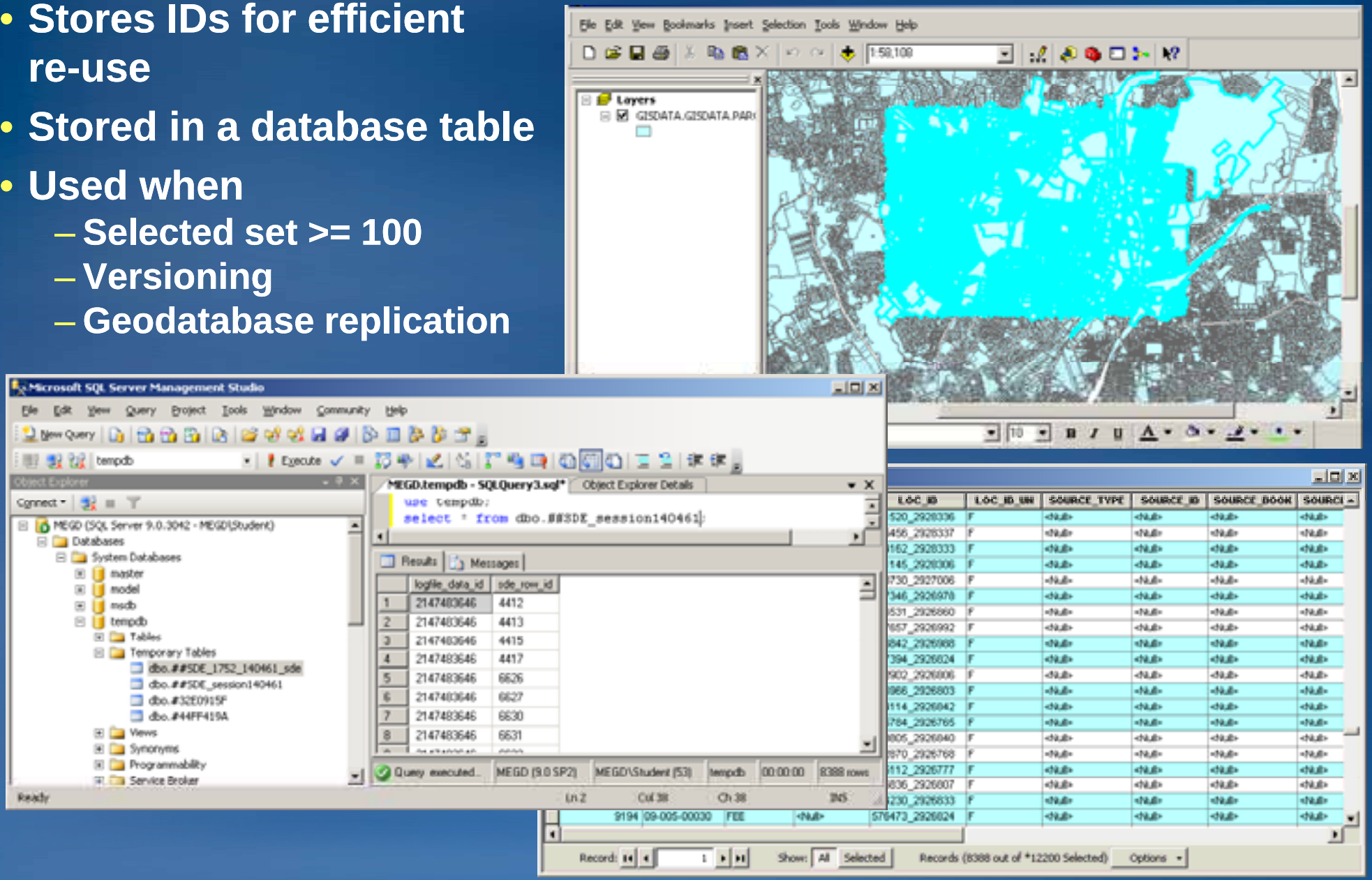
```
C:\sdeconfig -o alter -v MAXBLOBSIZE=2000000 -i esri_ora

ArcSDE 9.2 for Oracle10g
SDE Server Configuration Tool    Administration Utility
-------------------------------------------------
Alter SERVER_CONFIG Table. Are you sure? (Y/N): y
        Successfully altered SERVER_CONFIG Table.
```

AUTH_KEY
CONNECTIONS
RASTERBUFSIZE
MINBUFSIZE
MAXBUFSIZE
MINBUFOBJECTS
MAXTIMEDIFF
TEMP
MAXBLOBSIZE
BLOBMEM
AUTOCOMMIT
MAXINITIALFEATS
MAXDISTINCT
SHAPEPTSBUFSIZE
ATTRBUFSIZE
MAXARRAYSIZE
MAXARRAYBYTES
STREAMPOOLSIZE
STATECACHING
TCPKEEPALIVE
READONLY
STATUS
DEFAULTPRECISION
TLMINTERVAL
STATEAUTOLOCKING
LAYERAUTOLOCKING
INT64TYPES
MAXSTANDALONELOGS
ALLOWSESSIONLOGFILE
LOGFILEPOOLSIZE
HOLDLOGPOOLTABLES

ArcSDE log files?

- References a collection of IDs (list of selected records)
- Stores IDs for efficient re-use
- Stored in a database table
- Used when
  - Selected set >= 100
  - Versioning
  - Geodatabase replication



Types of ArcSDE log files

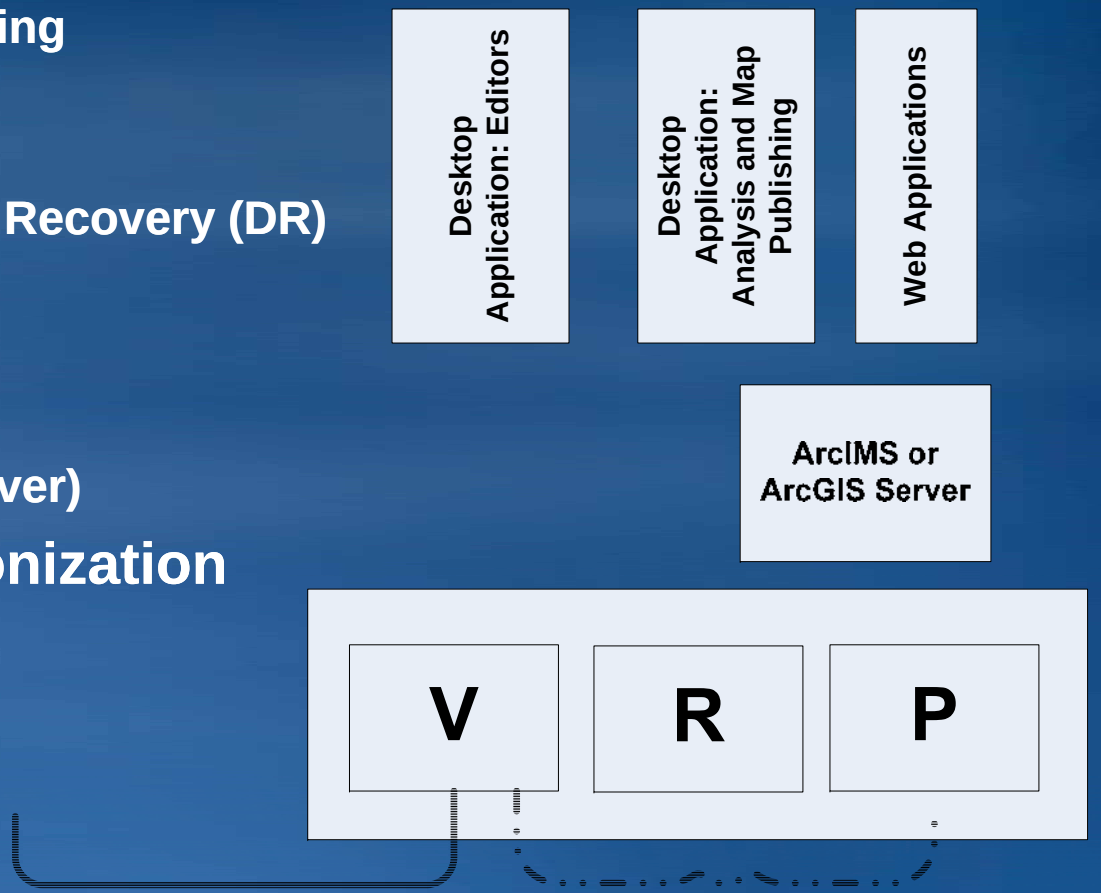
1. **Shared log file tables**
   - User-owned log file tables shared by all sessions
   - Default architecture
   - Note: Similar to 8.x log file architecture
2. **Session log file tables**
   - Log file table for each connection
   - SDE-owned pool or user-owned table
3. **Stand-alone log file tables**
   - Log file table for each layer selection
   - SDE-owned pool or user-owned table

Design: Log File Recommendations

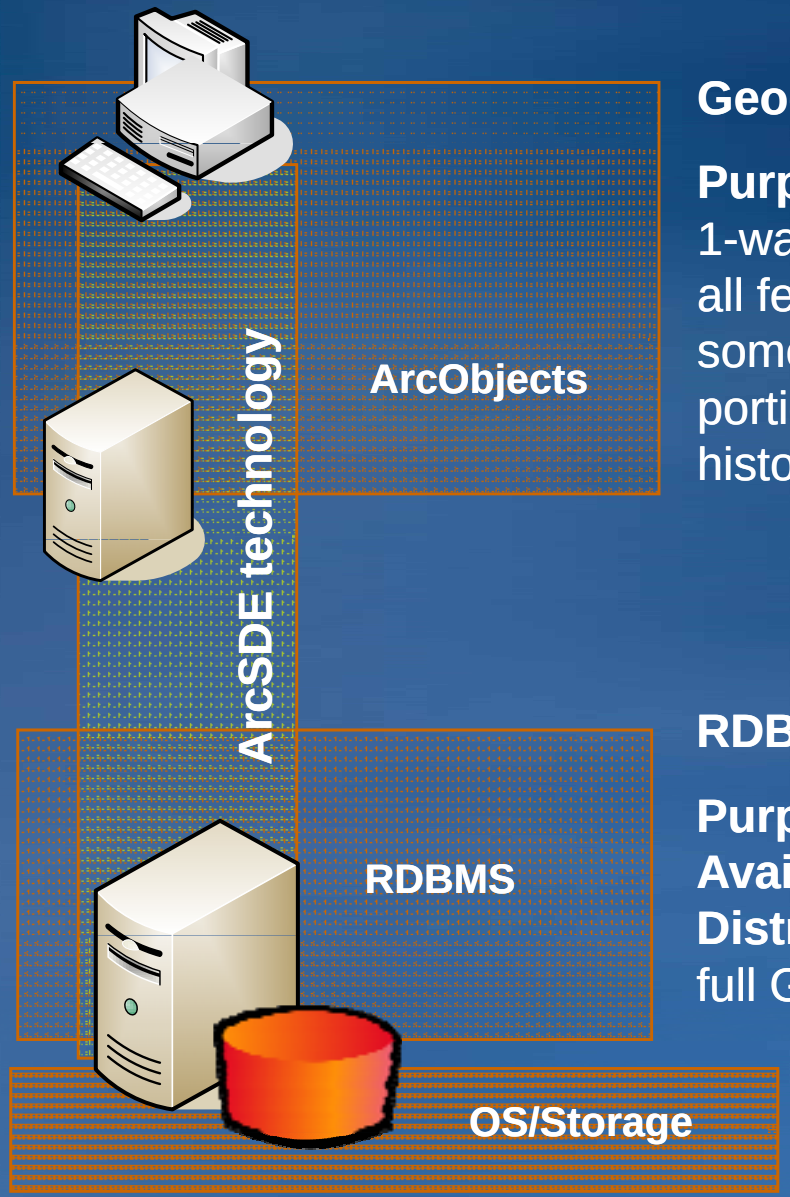
- Use default architecture for RDBMS
  - Shared for Oracle (IDs removed upon unselect versus disconnect)
    - Global temporary tables – see ESRI KB article 32161)
  - Session for SQL Server
  - see recommendations for other RDBMS
- Use session if multiple users have same login
  - Default for SQL Server (session/tempdb – minimizes logging)
  - Avoids table contention (e.g., many users connecting with a single login)
- Use pool if users are prohibited from creating objects
  - If pool unavailable, ArcSDE will attempt a user-owned table
    - Requires CREATE permissions
    - Error return with insufficient permissions

Design: Geodatabase Architecture

- Single vs. Multiple Geodatabases
  - Uses/Requirements
    - Vector and Raster, Editing and Publishing, Production and Development, Departmental
    - Performance – use specific tuning
    - Management/Administration (e.g. locking issues)
    - High Availability (HA), Disaster Recovery (DR)
  - Implementation
    - Multiple Instances (e.g. Oracle)
    - Multiple Databases or Named Instances (e.g. SQL Server)
- Data Distribution and Synchronization
  - Geodatabase Replication
  - Export/Import
  - Database Cloning or Replication



Design: Data Distribution and Replication



Geodatabase Replication

Purpose: Data Distribution
1-way, 2-way
all feature classes
some feature classes
portions of feature classes
history

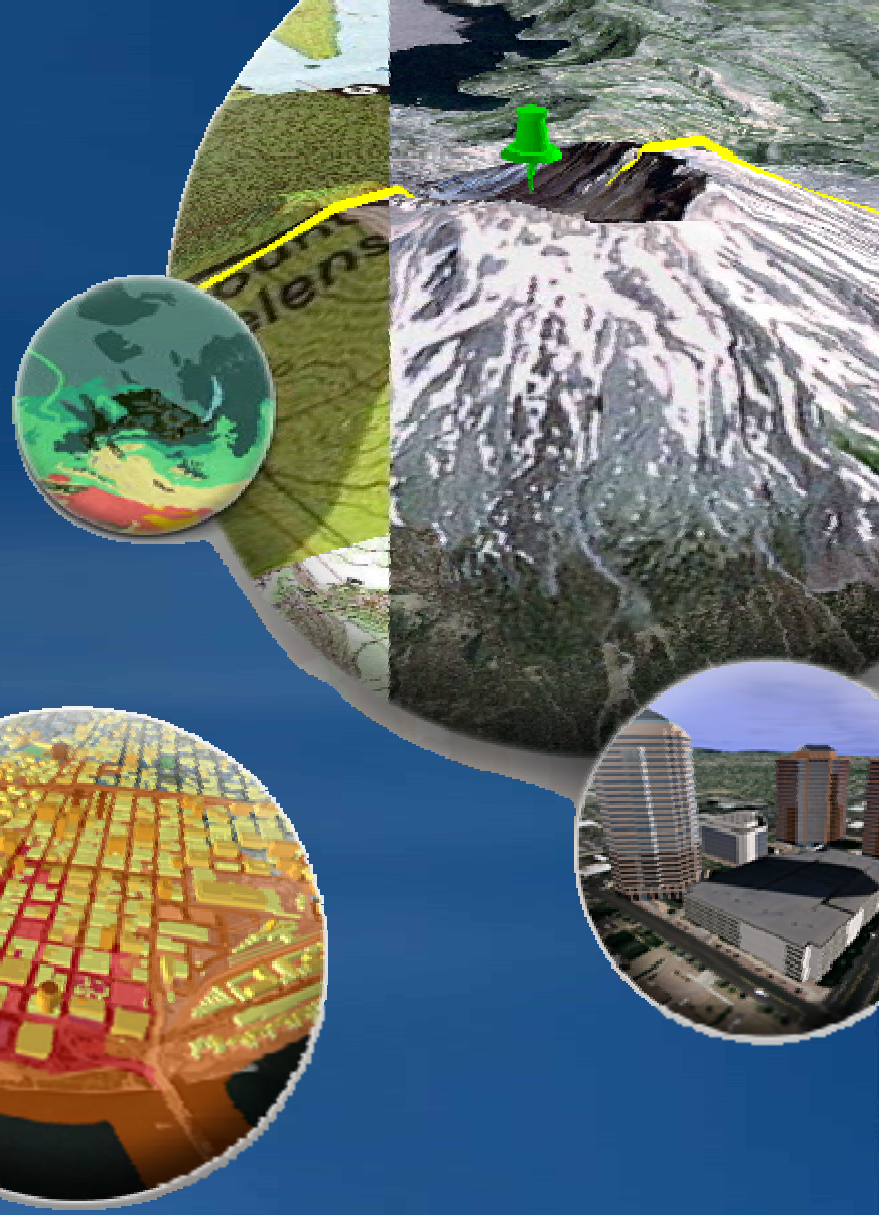
RDBMS/Storage Replication

Purpose: HA (High Availability), Data Distribution
full Geodatabase

Section 2:
Access and Data Management Best Practices

Feature Classes

- Complex geometries with high number of vertices (> 100,000)
  - split where possible
- Generalization
  - group spatially
- Denormalize geodatabase
  - combine feature classes where possible
- Data Size
  - All data impacts performance
    - Attributes
    - Geometry Storage
- Spatial Datatype (e.g. ST_Geometry in Oracle GDB – 9.3.1 SP1 secondary filtering)

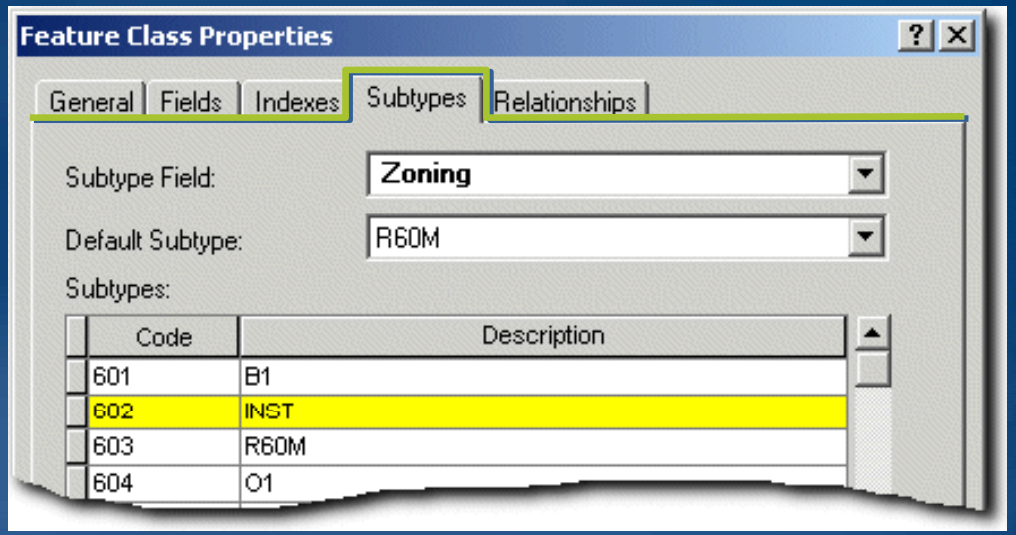


Feature Datasets

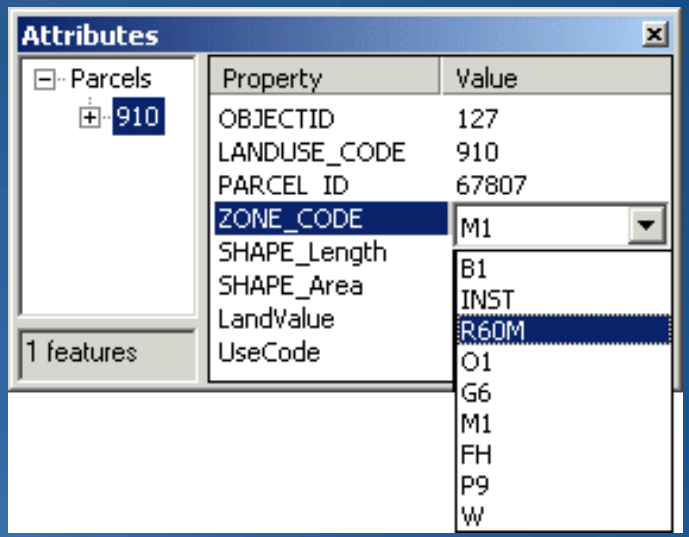
- Designed to ensure spatial coincidence among classes
  - Required for many types of behavior
    - Geometric networks, topologies, and so on
- Considerations for multiuser geodatabase design
  - All feature classes in a feature dataset are instantiated
  - Privileges are granted/revoked for all classes
  - Registering as versioned occurs at dataset level
  - Locks can apply to all feature classes (can be an issue)
    - Spatial Views, Separate Editing and Publishing geodatabases, etc… can sometimes help
- Do not use for organizational purposes

Subtypes and Domains

- Subtype: Saved classification
  - Classifications share behavior
  - Maintains data integrity
- Subtypes & Performance
  - Better than storing data in many feature classes
    - A query is generated for each feature class displayed
  - Automatically symbolized by subtype in ArcMap

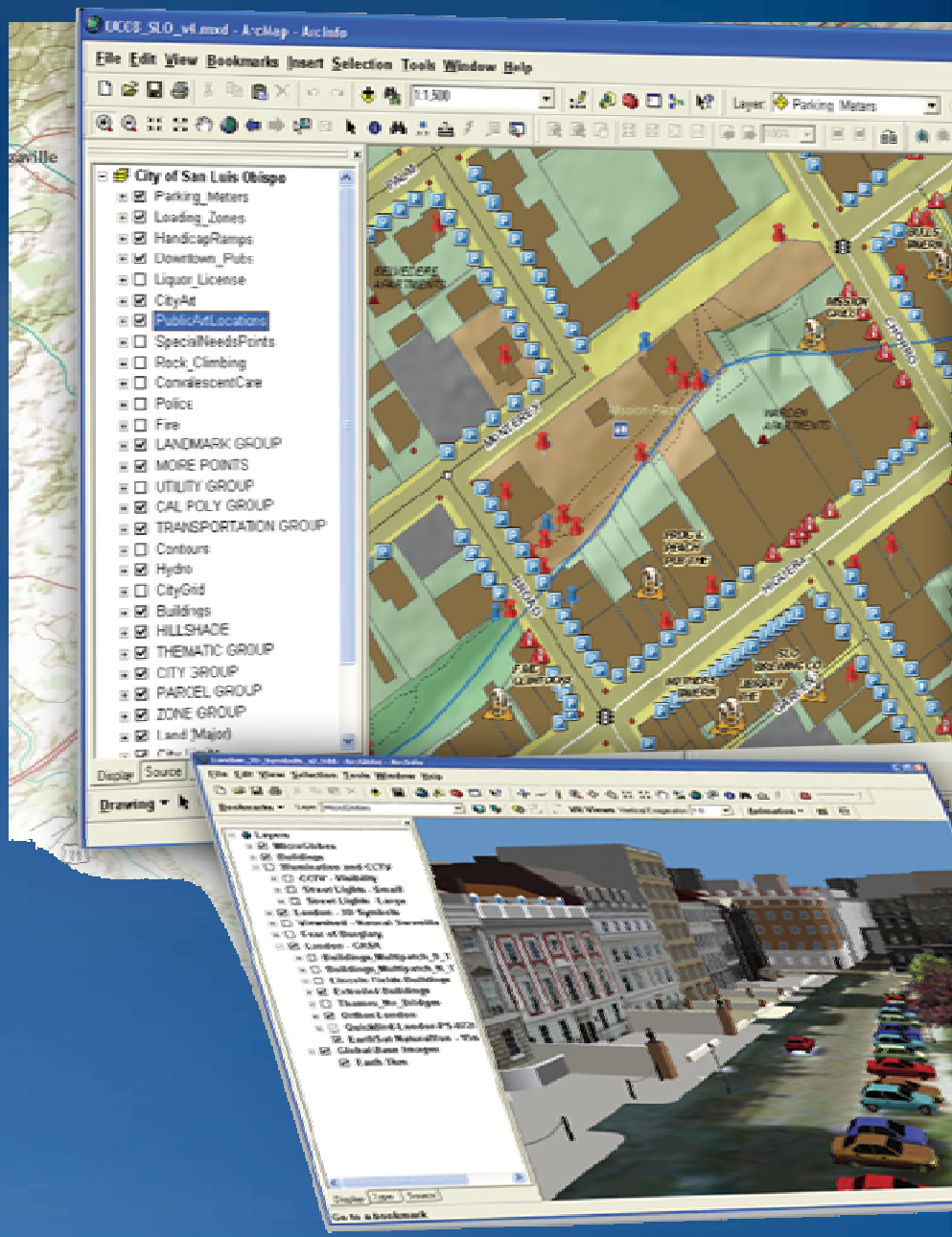


- Domains: Set of legal values for a field's attribute
  - Range: Minimum/maximum values
  - Coded value: Codes and descriptions
- Domain & Performance
  - Range: Negligible performance impact
    - Behavior on demand
  - Coded value: Minor performance impact during edit session
    - Need to generate list of attributes

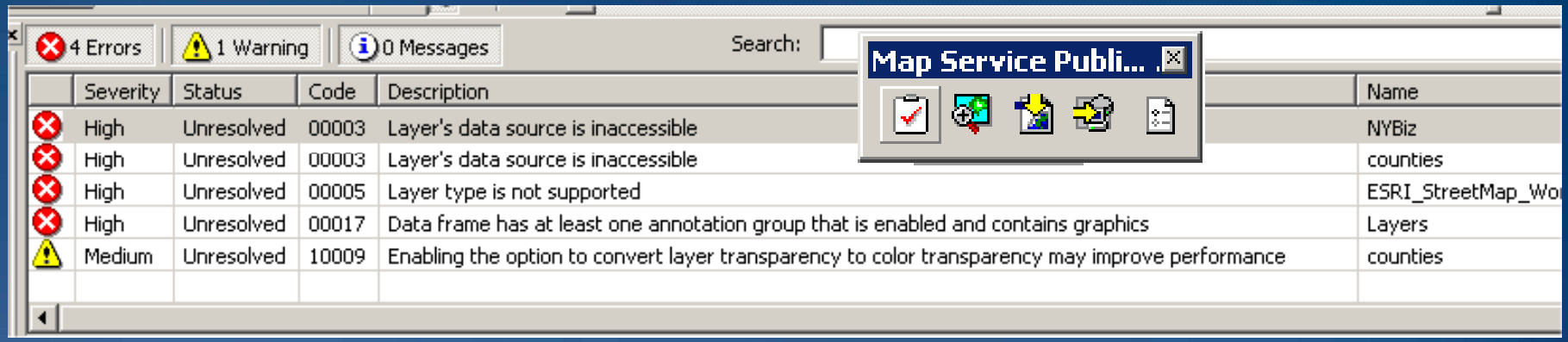


Labeling, Symbols, Sub-Queries and Renderers

- **Impact of labeling and symbols**
  - Causes a 2nd SQL query on every layer (feature, then label attribute)
  - Enable Map Cache or use Annotation
- **Impact of Sub-Queries**
  - Can reduce amount of information returned by filtering attributes
  - Can cause extra load on the database if columns not indexed
- **Impact of Renderer**
  - More complexity in rendering and symbology increase CPU load, especially in Citrix/Terminal server architectures.
  - Switch to more appropriate renderers.



Data Access: Tools for Map Documents



- map document performance analysis tool at 9.3.1
- New .msd document @ 9.3.1
  - Map Service Definition file for 9.3.1 ArcGIS Server fast drawing engine
- MxdPerfStat (arcscripts.esri.com)
  - Check mxd performance

Item	Layer Name	At Scale	Refresh Time (sec)	Recommendations	Features	Vertices	Labeling	Geography Phase (sec)	Graphics Phase (sec)	Cursor Phase (sec)	DBMS CPU	DBMS LIO	DBMS PIO	DBMS
2	STUDENT.parcel_1	50,000	8.22	set scale dependency; run DBMS trace;	30,628	153,140	True	4.27	3.84	5.74	2.37	69,995		esriDBMS_Oracle

Client Options: ArcMap best practices for users

- Avoid full display
  - ArcMap magnifier and overview windows
  - Scale dependencies
  - Use spatial bookmarks
- Set selectable layers
- Keep table of contents and symbology simple
- Use keep only matching records option with joins
- Use map cache

Client Options: Caching

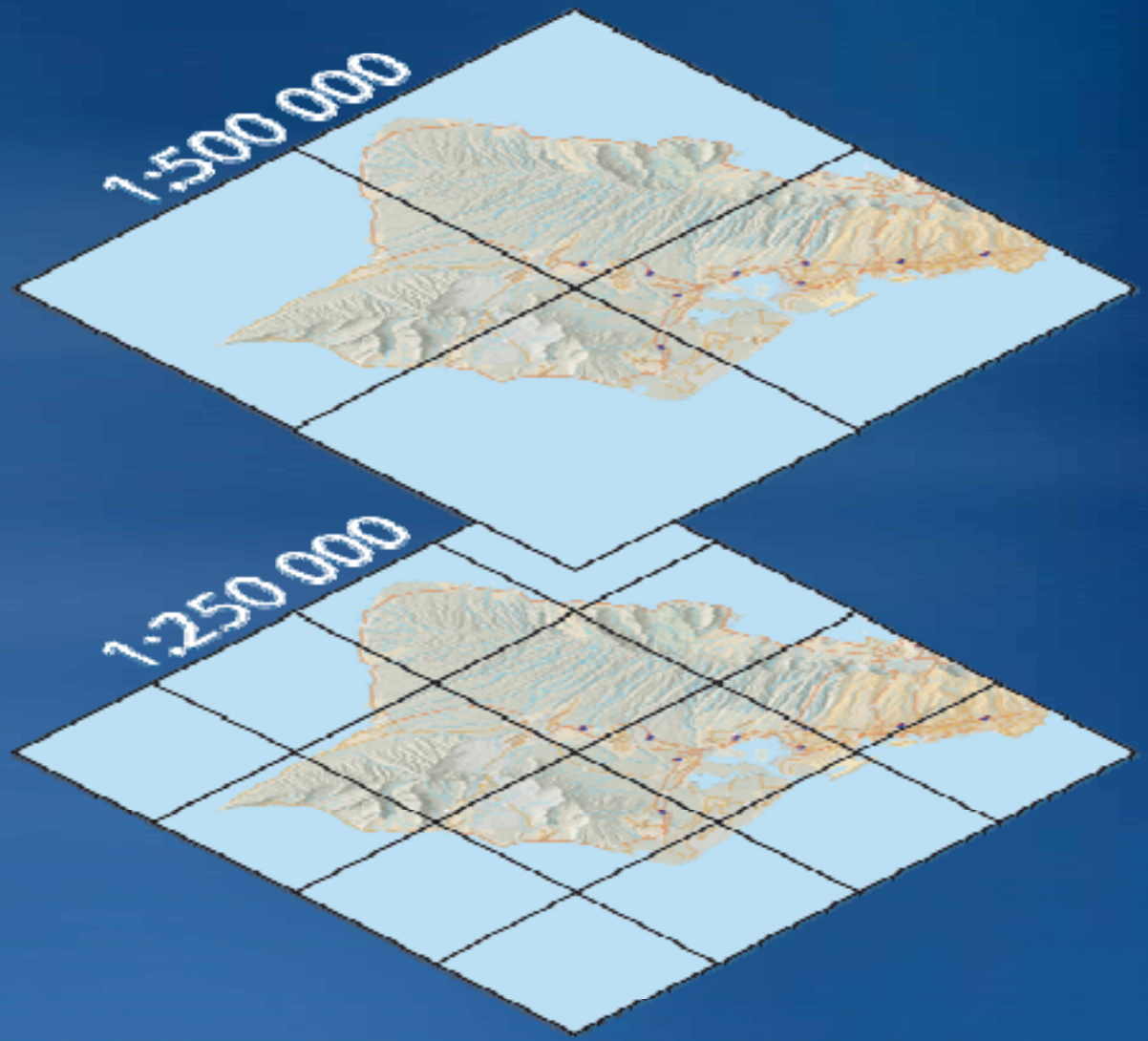
- ArcGIS Desktop Map cache
- ArcGIS Server Map Cache

ArcGIS Desktop Map Cache

- Client side caching of feature values over a given spatial extent
- Can speed up queries
  - Reduces roundtrips to the database
- When to use?
  - If making many spatial queries within a common extent
  - When you anticipate working with several features within a certain geographic area, e.g. editing within a larger area
  - When editing non-simple Geodatabase features, e.g. Geometric Networks
  - When editing and snapping enabled. Each snap requires a round-trip(s) to the database unless there is a cache
- Do NOT use for non-versioned editing (short-transaction)

ArcGIS Server Map Cache

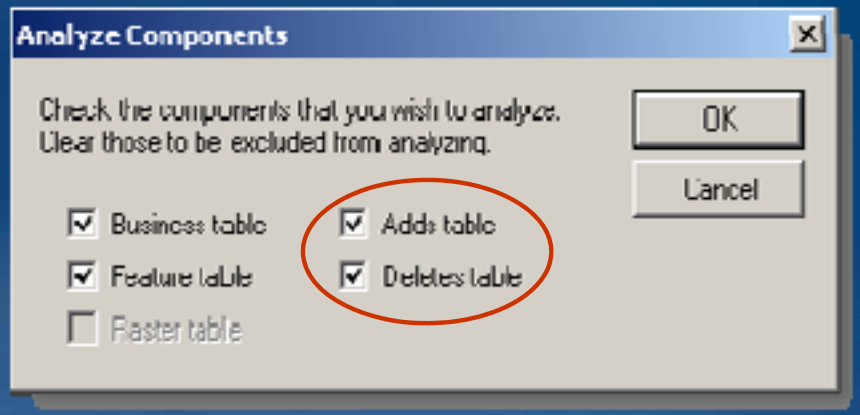
- Tiles pre-rendered at fixed scales
- Rapid display of static base maps
- Richer symbols and more information



Section 3:
Maintenance Best Practices

Maintaining DBMS statistics

- ArcSDE does not maintain statistics; DBA responsible
  - Describe data for database optimizer
  - Critical for maintaining performance



- Keep up-to-date statistics
  - Depends on editing activity
  - Before and after database compress
  - It is better to have “over-reported” (or none) then “under-reported” statistics

Which tables need DBMS statistics updated?

- Base tables
  - Fairly static (unless compressing to base)
  - Therefore, typically do not need to update statistics frequently
- A- and D-tables (can) require frequent updates
- F- and S-tables (can) require frequent updates
- STATE_LINEAGES needs to have good statistics
  - Update frequently
- Other versioned repository tables that are frequently queried:
  - States
  - Mvtables_modified
- Raster tables are generally static

Maintaining indexes

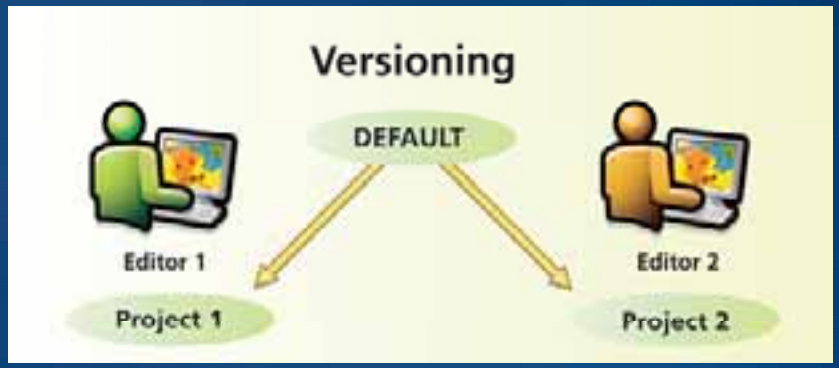
- Inserting and deleting causes entries to be scattered
  - Insertions introduce non-contiguous entries
  - Deletions cause skewed index
- Poor indexes can lead to increased I/O
- ArcSDE does not coalesce or rebuild after compress
  - Rebuild, coalesce indexes, maintain index statistics
- For versioned databases
  - Periodically rebuild delta table indexes
- Automate Index Management
  - Each RDBMS offers functionality to automate Index management via “jobs/tasks”
  - ArcGIS geoprocessing tools can also be used (and automated)

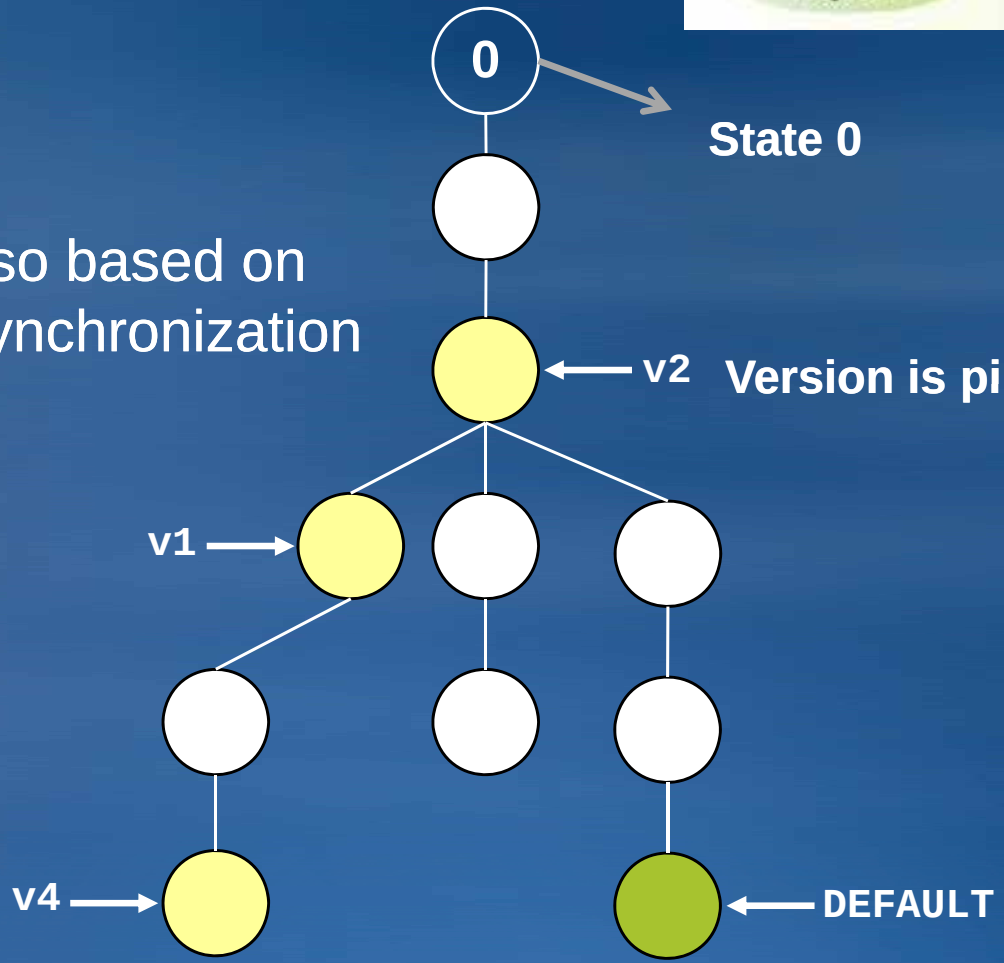
Versioned Editing Managment

- Reasons performance can degrade over time
  - Workflow can often generate unnecessary versions
    - Versions may just “hang-out” on the tree
    - Prevents state tree from being fully compressed
  - Response time increases with volume of states
    - In production, number of rows can increase significantly
    - Table growth is a function of edits
  - Performance lags introduced with stale statistics
    - Database might choose a sub-optimal execution plan

Versioned Editing Performance

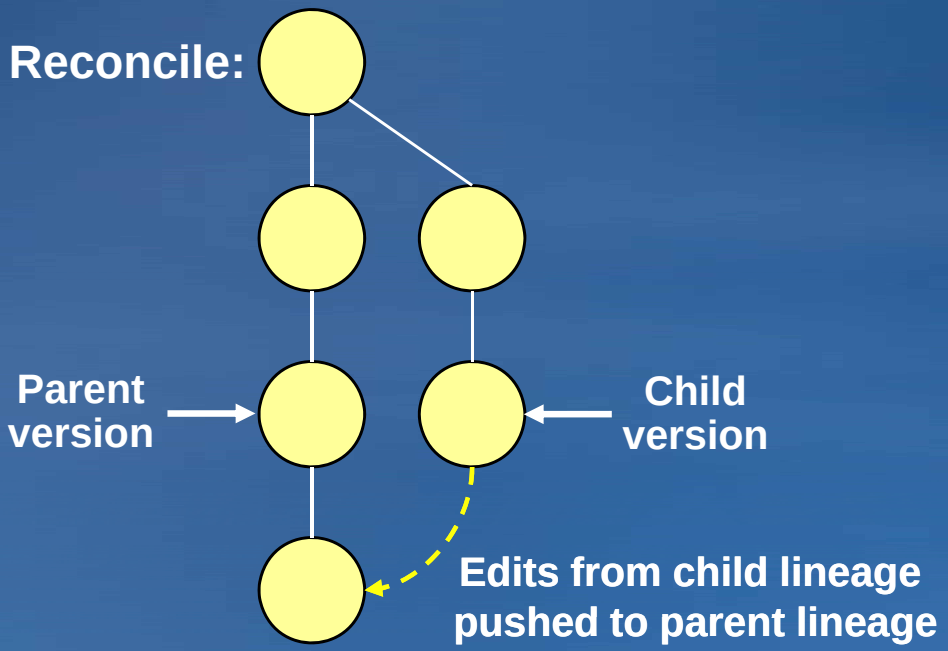
- Manage your version workflow to reduce depth of state tree
  - Reconcile
  - Post
  - Compress
- Replication also based on versioning - synchronization

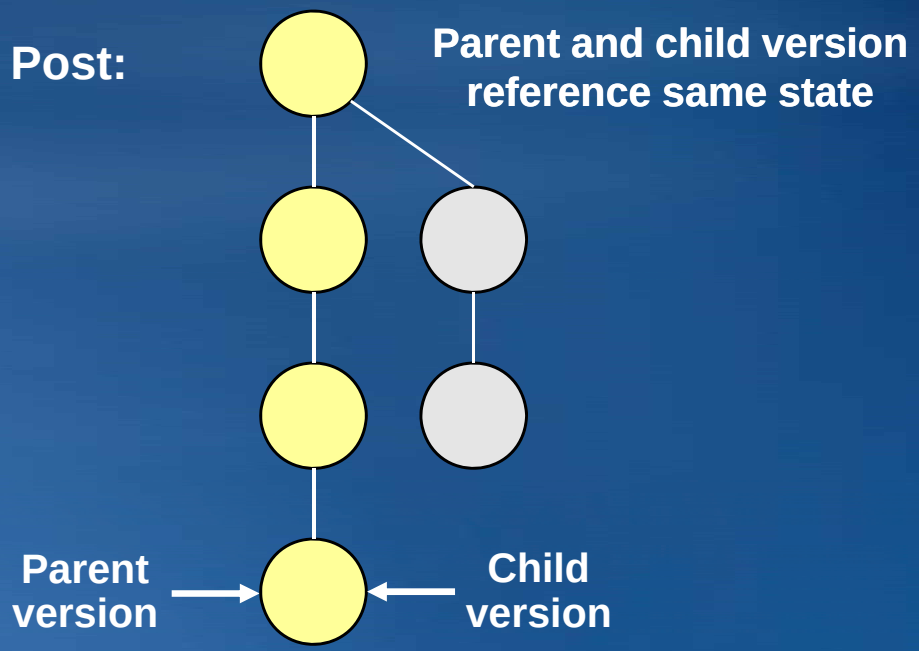




What is reconcile and post?

- Reconcile:
  - Pushes edits from one branch of the state tree to target branch
  - Searches for conflicts
- Post:
  - Saves the new state of the version





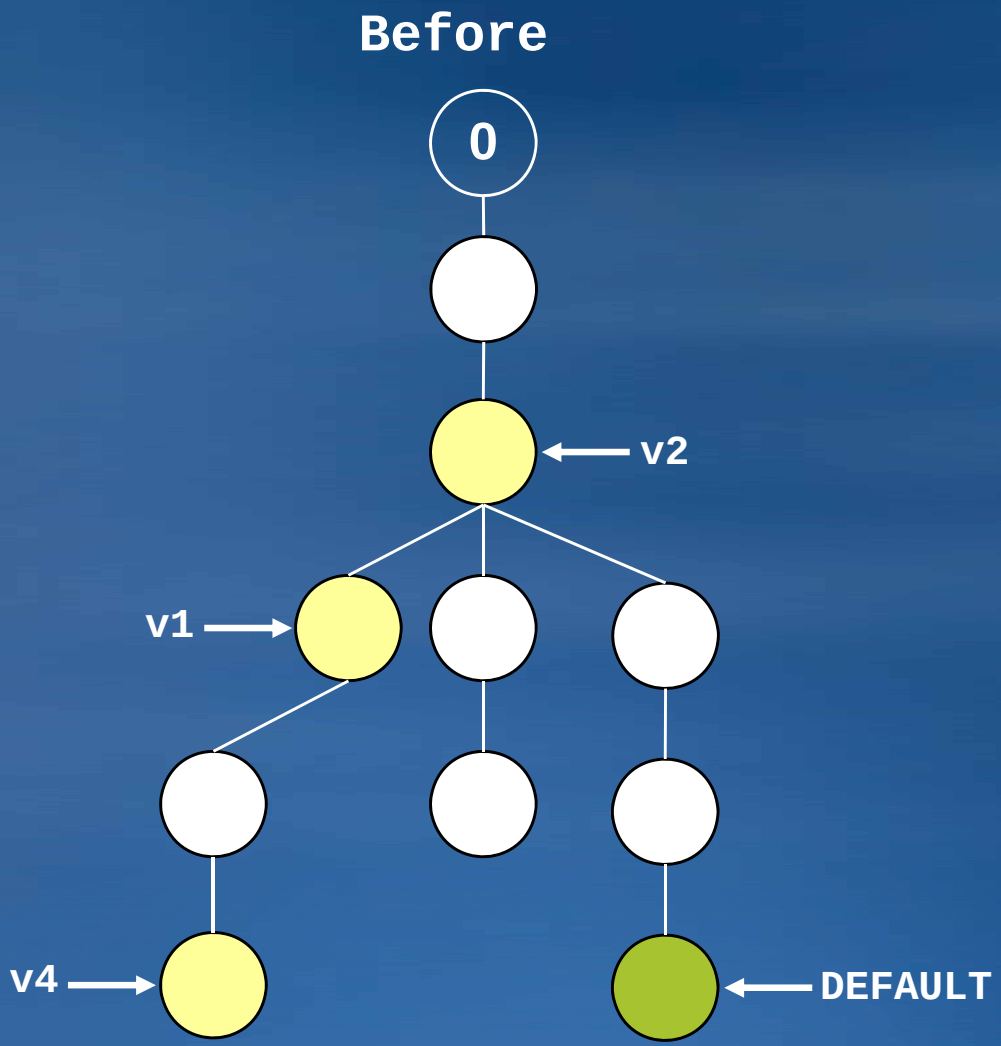
Automation of Reconcile / Post

- **KB36809: Report the recommended reconcile order using SQL in SQL Server**
- **KB35735: Report the recommended reconcile order using SQL in Oracle**
- **Use GP Tools and script to python**
- **AO sample code (old samples): Reconcile SDE versions in batch mode**

http://edndoc.esri.com can search for "*Reconcile SDE versions in batch mode*" – (*Note: some of these are 9.1 VB6 samples and might throw a license error, check discussion forum for more information on how to recompile*)

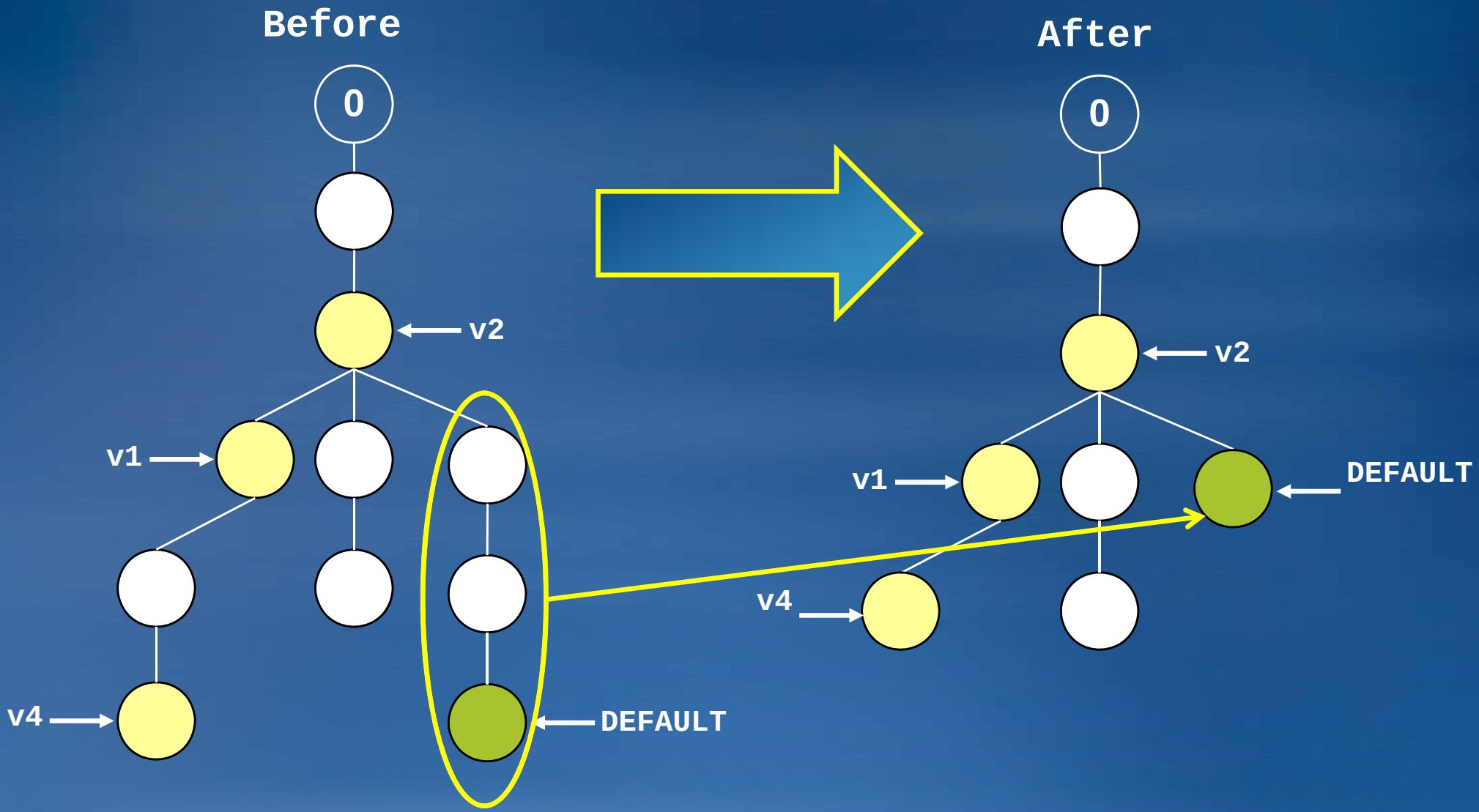
Importance of compressing the database

- Moves edits from multiple states into one state on a shared branch



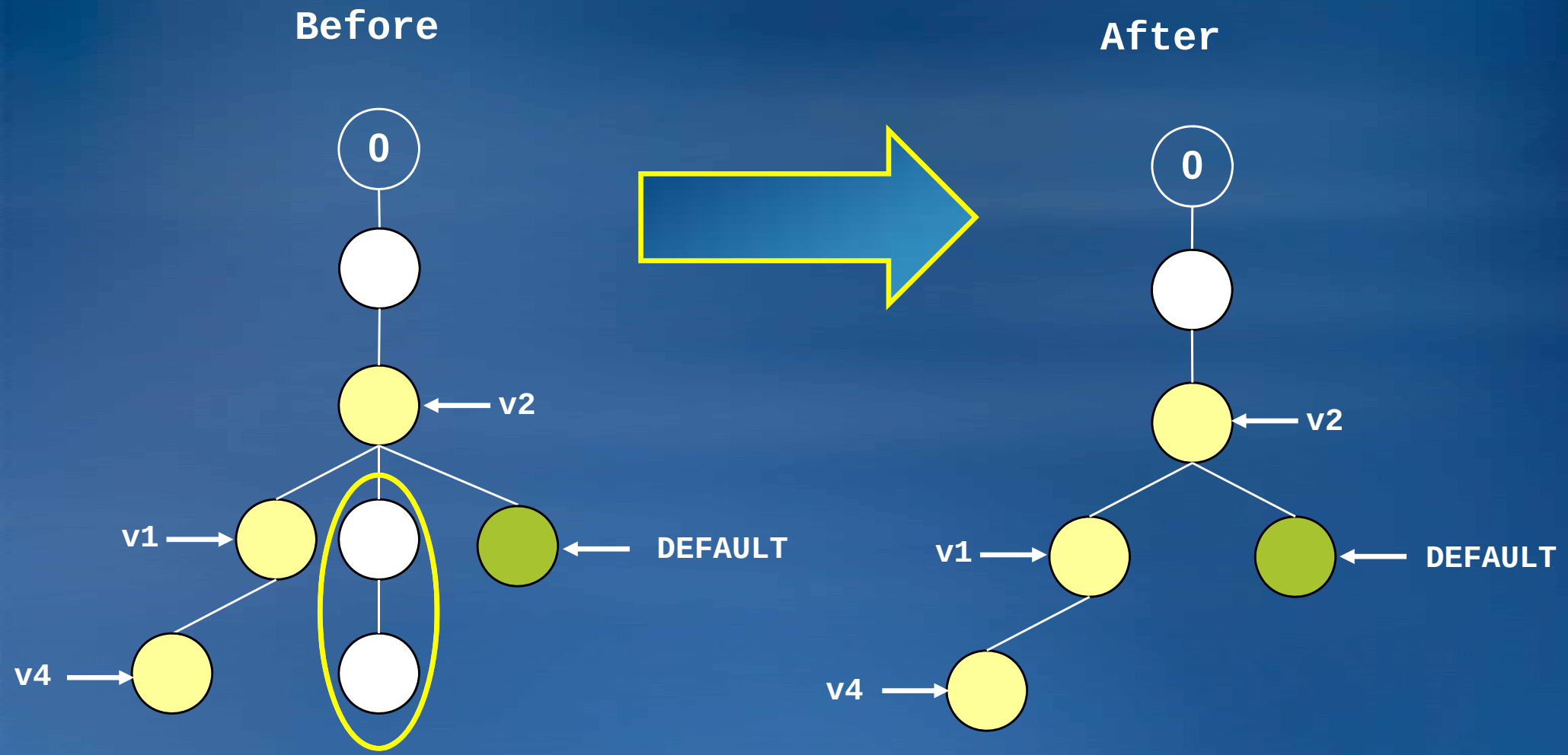
Importance of compressing the database

- Moves edits from multiple states into one state on a shared branch



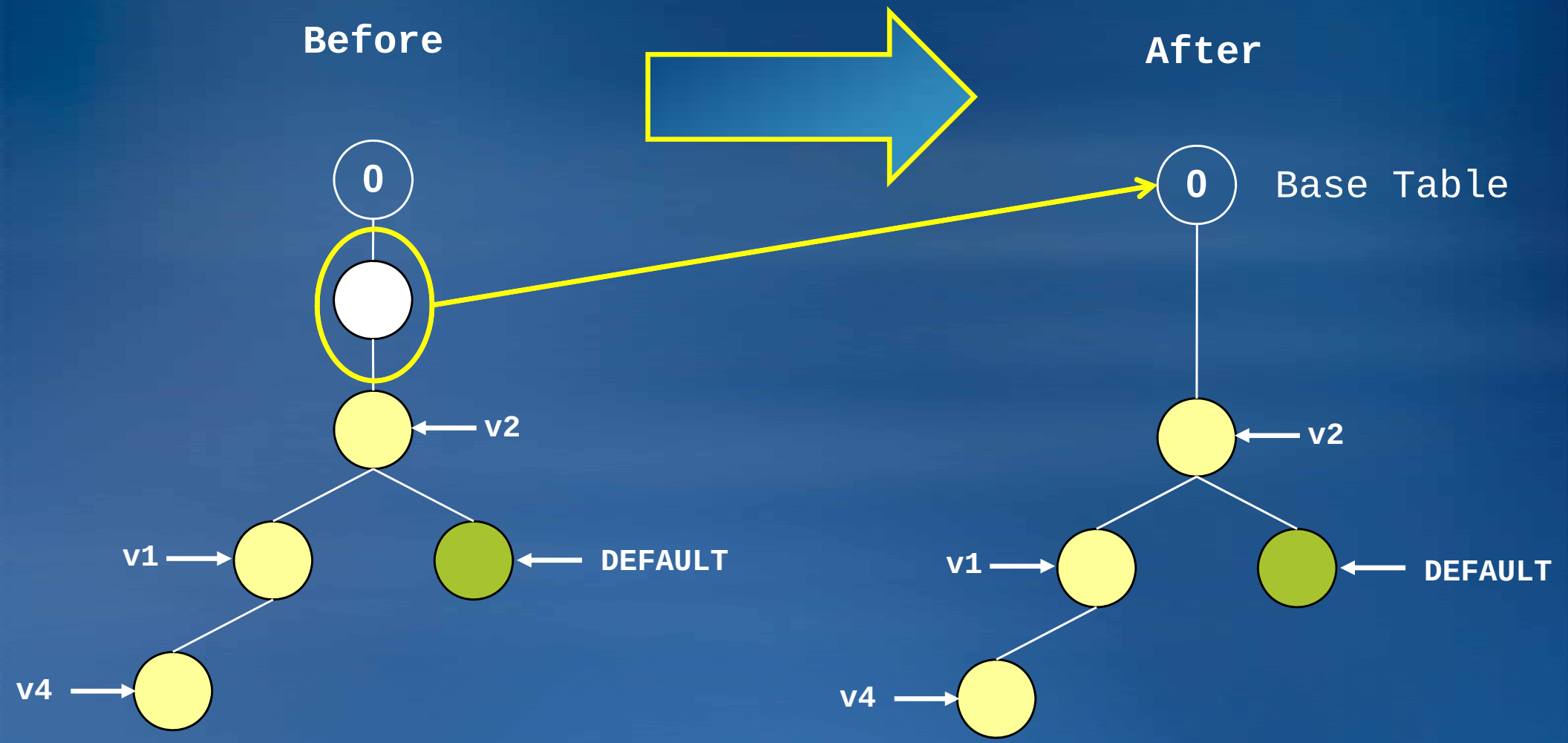
Importance of compressing the database

- Removes unreferenced states



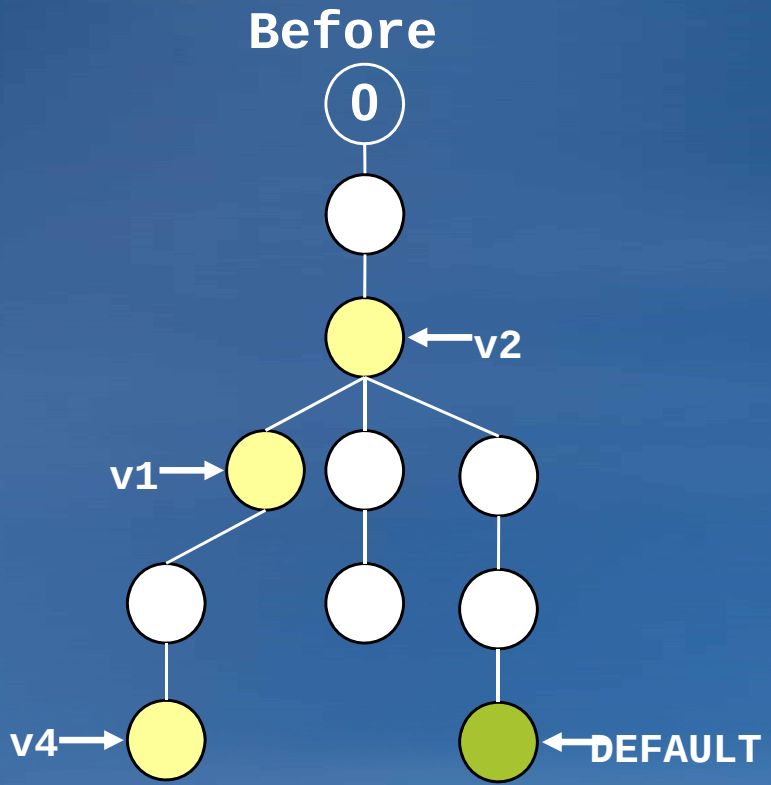
Importance of compressing the database

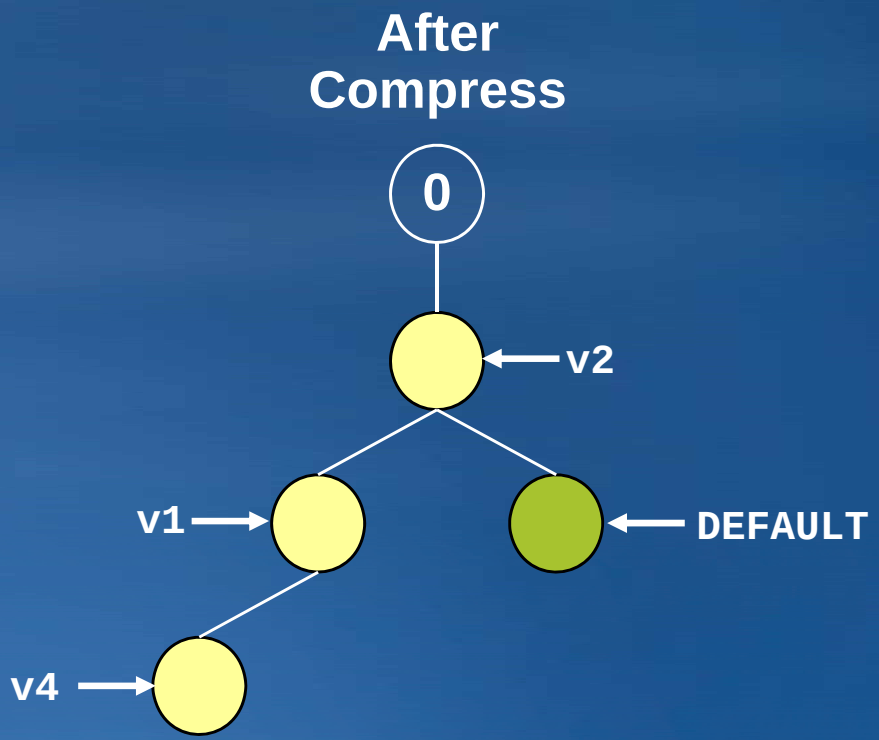
– Moves common rows from delta tables into base tables



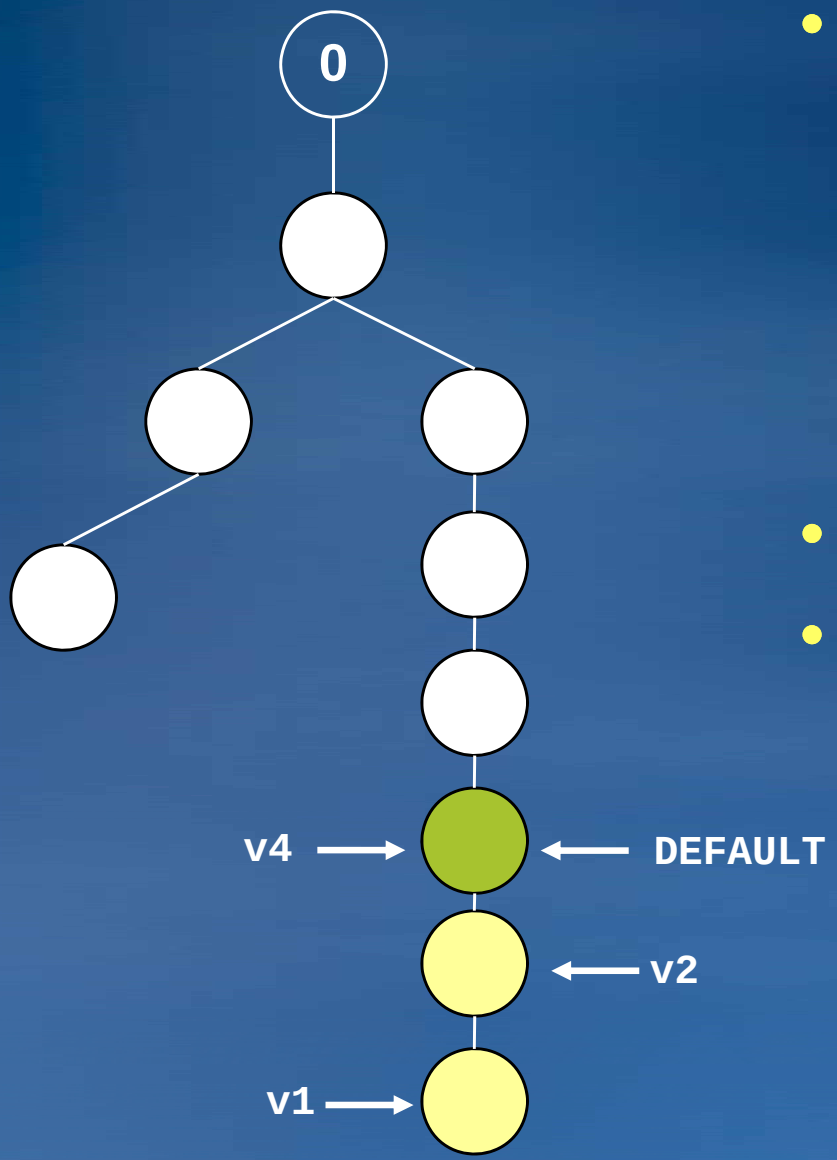
Importance of compressing the database

- Maintains Performance and Health
  - Moves common rows from delta tables into base tables
  - Reduces depth of state tree
    - Removes redundant rows
    - Removes unreferenced states (save points)





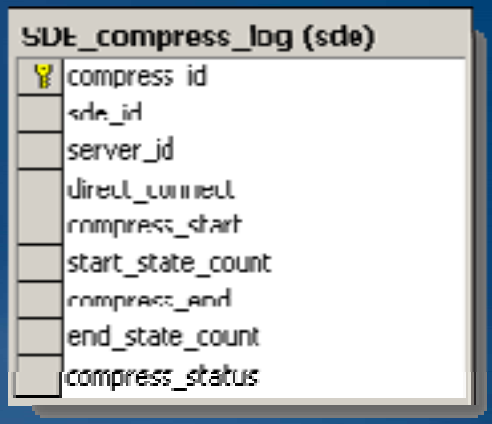
Compressing all rows to the base table



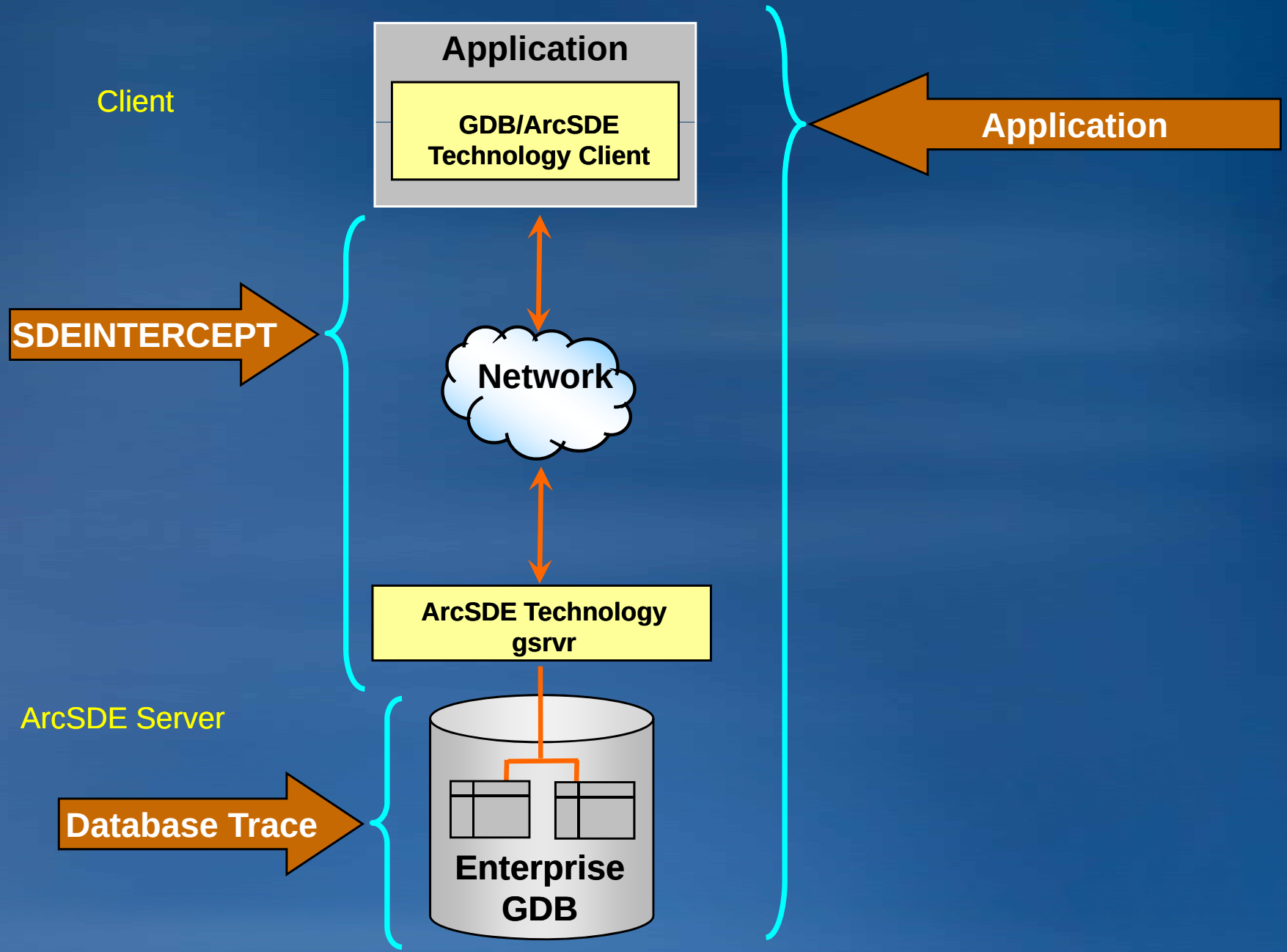
- Maximum benefit: Compress all rows to base tables
  1) Reconcile and Post all versions to DEFAULT
  2) Must Reconcile a 2nd time
  3) Compress
- State tree pointing at zero
- Not always an option
  - Workflow requirements
  - Also an option to delete versions at step 2 instead of second reconcile.

How often should you compress?

- Depends on the amount of editing activity
- Not unreasonable to compress every night or during the day
  - Routine maintenance for highly edited databases
- Compress at least once a week
  - Medium to low volume of edits
- **`compress_log`** provides information on compress
  - **`start_time and end_time`**
  - Number of states compressed



Log Availability



Log File collection:

- Important to collect a set while the system is operating correctly.
- Are available for the entire stack –
  - ArcGIS Desktop (client)/Arcobjects/ArcSDE
  - ArcGIS Server
  - RDBMS
  - OS
- Useless without knowing the context of what was being logged.

Logs and Tracing

What is SDE Intercept

- Built-in ArcSDE functionality
- Logs the ArcSDE client calls to the ArcSDE server
- Useful to profile how many and what type of calls are made

Purposes

- Check performance
- Establish performance benchmark (under typical workload)
- Troubleshoot Errors

Enabling SDE Intercept

- Client side - only client session
- Server side - all sessions
- Based on connection to the SDE geodatabase
- set SDEINTERCEPTLOC=<file location>
- set SDEINTERCEPT=<flags>

Example: SDEINTERCEPT Output RDBMS problem

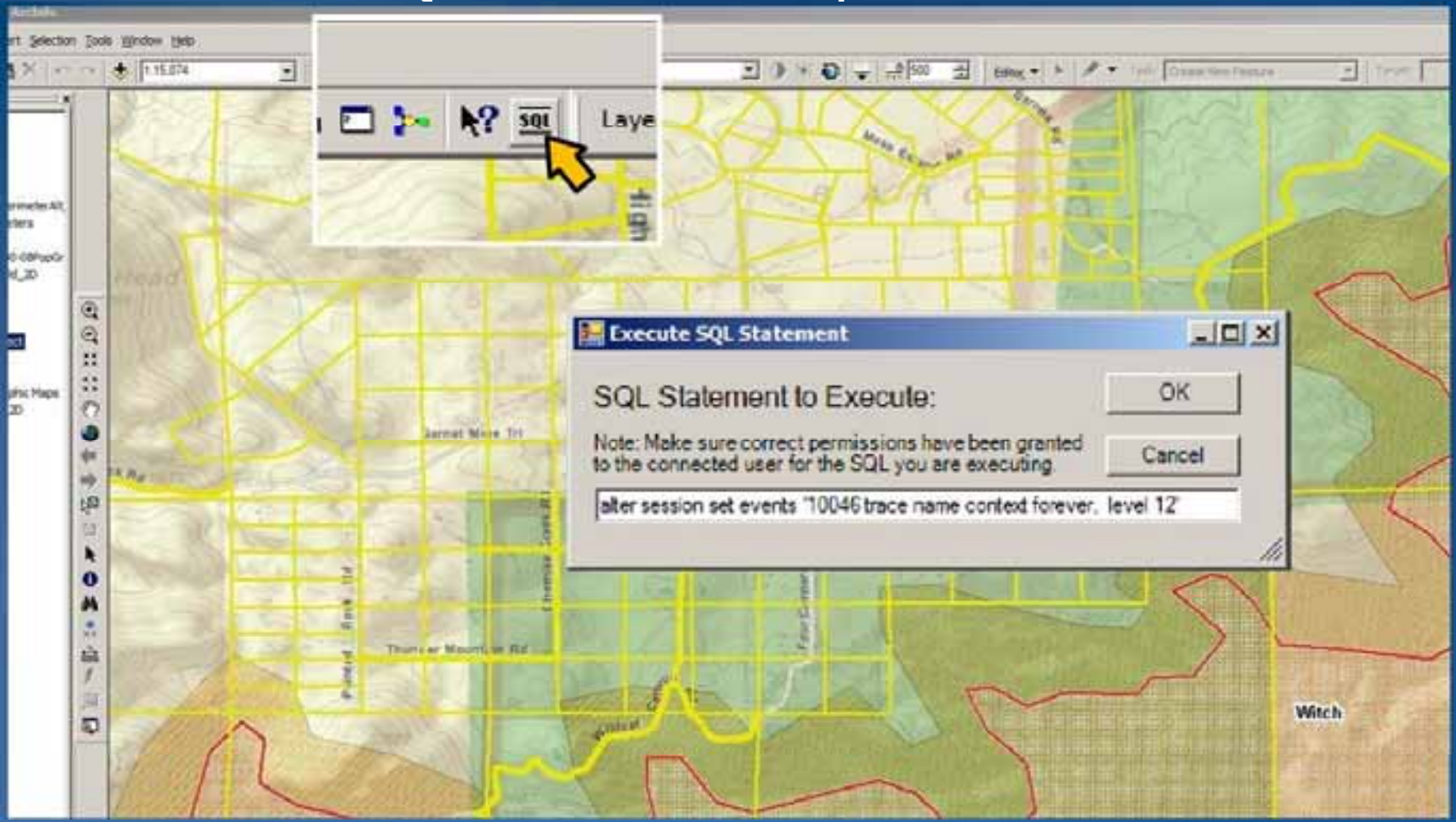
SDEINTERCEPT

```
========================================
[W 18:01:06] Command:     ExecuteSpatialQuery
[W 18:01:06] Long:          1
[R 18:01:06] Long:          0
========================================
[W 18:01:06] Command:     NextBuffer
[W 18:02:44] Long:          1
[R 18:02:44] Long:          0
[R 18:02:44] Long:          15312
[R 18:02:44] Long:          353
[R 18:02:44] Short:         1
[R 18:02:44] Long:          1
[R 18:02:44] Long:          0
[R 18:02:44] Block:
  BufferInfo:     [25/15312]  Address@0xc9c0000
  BufferInHex:    "020039AA4E0002000000004001B0000001301000001000000 9A..."
========================================
```

Example: enable DBMS trace

Database Trace: from ArcMap

- **IWorkspace.ExecuteSQL GUI tool (Custom add-in command for ArcGIS)**
  - **Available at http://resources.esri.com in the ArcGIS Desktop DotNet Code Gallery**
  - **Search for "ExecuteSQL Command for ArcMap"**



NOTE: User must have database specific priveledges.

Example: DBMS trace output on Oracle

Database Trace: e.g. find Streets function

```
SELECT  1 SHAPE, DETAILED_STREETS_FOR_CA.OBJECTID,
  DETAILED_STREETS_FOR_CA.SHAPE.points,DETAILED_STREETS_FOR_CA.SHAPE.numpts,
  DETAILED_STREETS_FOR_CA.SHAPE.entity,DETAILED_STREETS_FOR_CA.SHAPE.minx,
  DETAILED_STREETS_FOR_CA.SHAPE.miny,DETAILED_STREETS_FOR_CA.SHAPE.maxx,
  DETAILED_STREETS_FOR_CA.SHAPE.maxy,DETAILED_STREETS_FOR_CA.rowid
FROM
  ROB.DETAILED_STREETS_FOR_CA DETAILED_STREETS_FOR_CA WHERE
  SDE.ST_EnvIntersects(DETAILED_STREETS_FOR_CA.SHAPE,:1,:2,:3,:4) = 1


call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        0      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        4     73.15      98.17     142167     142566          0         353
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        5     73.15      98.17     142167     142566          0         353

Misses in library cache during parse: 0
Optimizer mode: ALL_ROWS
Parsing user id: 82  (ROB)

Rows     Execution Plan
-------  ---------------------------------------------------
      0  SELECT STATEMENT   MODE: ALL_ROWS
      0   TABLE ACCESS   MODE: ANALYZED (FULL) OF
            DETAILED_STREETS_FOR_CA (TABLE)
```

Add Spatial Index

- Full table scan, no spatial index
- Need to add spatial index

Example: SDEINTERCEPT Output RDBMS problem fixed

SDEINTERCEPT

```
=========================================
[W 19:33:22] Command:      ExecuteSpatialQuery
[W 19:33:22] Long:         1
[R 19:33:22] Long:         0
=========================================
[W 19:33:22] Command:      NextBuffer
[W 19:33:22] Long:         1
[R 19:33:22] Long:         0
[R 19:33:22] Long:         15312
[R 19:33:22] Long:         353
[R 19:33:22] Short:        -1
[R 19:33:22] Long:         1
[R 19:33:22] Long:         0
[R 19:33:22] Block:
  BufferInfo:   [25/15312]  Address@0xc9c0000
  BufferInHex:
        "020093AA4E000200000004001900000011010000100000084..."
=========================================
```

Example: DBMS trace output on Oracle

Database Trace: e.g. findParcels function

```
SELECT  1 SHAPE, DETAILED_STREETS_FOR_CA.OBJECTID,
  DETAILED_STREETS_FOR_CA.SHAPE.points,DETAILED_STREETS_FOR_CA.SHAPE.numpts,
  DETAILED_STREETS_FOR_CA.SHAPE.entity,DETAILED_STREETS_FOR_CA.SHAPE.minx,
  DETAILED_STREETS_FOR_CA.SHAPE.miny,DETAILED_STREETS_FOR_CA.SHAPE.maxx,
  DETAILED_STREETS_FOR_CA.SHAPE.maxy,DETAILED_STREETS_FOR_CA.rowid
FROM
  ROB.DETAILED_STREETS_FOR_CA DETAILED_STREETS_FOR_CA WHERE
  SDE.ST_EnvIntersects(DETAILED_STREETS_FOR_CA.SHAPE,:1,:2,:3,:4) = 1


call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        0      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        4      0.00       0.00          0        304          0         353
------- ------  --
total        5      0.00       0.00          0        304          0         353


Rows     Execution Plan
-------  ---------------------------------------------------
      0  SELECT STATEMENT   MODE: ALL_ROWS
      0   TABLE ACCESS   MODE: ANALYZED (BY INDEX ROWID) OF
              'DETAILED_STREETS_FOR_CA' (TABLE)
      0    DOMAIN INDEX ((Sel: Default - No Stats)) OF 'A170_IX1' (INDEX
              (DOMAIN))
```

Example: How you can analyze a problem if you have both good and bad intercepts of RDBMS problem

Useful Measurments	Broken Intercept (no index example)	Good Intercept with Index
Query Start Time	18:01:06	19:33:22
Query Stop Time	18:02:44	19:33:22
Total elapsed time (calculated)	1 minute 38 seconds	Less than 1 second
Total number of rows returned	353	353

Example: SDEINTERCEPT Output Client problem

- Client issues show up as missing time between the completion of an operation.
- Must know what operation is being logged
  - Was there any user interaction
  - Automated process such as logon
- Client operations are normally separated by =====

Example: SDEINTERCEPT Output Client problem

SDEINTERCEPT of a connection in ArcCatalog

```
[R 13:35:57] Dynamic_Str:  "ROB.RESTAURANTS"
[R 13:35:57] Dynamic_Str:  "ROB.DETAILED_STREETS_FOR_CA"
=======================================
[W 13:36:04] Command:      StreamSetState
[W 13:36:04] Long:         1
[W 13:36:04] int64:        0
[W 13:36:04] int64:        -1
[W 13:36:04] Long:         0
[R 13:36:04] Long:         0
=======================================
[W 13:36:04] Command:      QueryWithInfo
[W 13:36:04] Long:         1
[W 13:36:04] Query Info:
   Num Columns:   16
   Columns:       "SDE.GDB_RelClasses.ID", "SDE.GDB_RelClasses.DatabaseName",
   "SDE.GDB_RelClasses.Owner", "SDE.GDB_RelClasses.Name",
   "SDE.GDB_RelClasses.OriginClassID", "SDE.GDB_RelClasses.DestClassID", <null>
```

Q&A